

**UNIVERSIDADE FEDERAL DO PARANÁ**  
**FRANCO BRANDELERO**

**Um Esquema de Alocação de Largura de Banda Dinâmica para  
Transmissão VoIP via Satélite**

**CURITIBA**  
**2010**

**FRANCO BRANDELERO**

**Um Esquema de Alocação de Largura de Banda Dinâmica para  
Transmissão VoIP via Satélite**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Engenharia Elétrica.

Programa de Pós-Graduação em Engenharia Elétrica, Departamento de Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná.

Orientador: Prof. Dr. Evelio Martín García Fernández

**CURITIBA  
2010**

UNIVERSIDADE FEDERAL DO PARANÁ  
SISTEMA DE BIBLIOTECAS  
BIBLIOTECA CENTRAL – COORDENAÇÃO DE PROCESSOS TÉCNICOS

- 
- B817e      Brandelero, Franco, 1981-  
Um esquema de alocação de largura de banda dinâmica para  
transmissão VoIP via satélite [manuscrito] / Franco Brandelero. –  
2010.  
92 f. : grafs., tabs. ; 30 cm.
- Impresso.  
Dissertação (mestrado) – Universidade Federal do Paraná, Setor de  
Tecnologia, Departamento de Engenharia Elétrica, Programa de Pós-  
graduação em Engenharia Elétrica, 2010.  
“Orientador: Prof. Dr. Evelio Martín García Fernández”.  
Bibliografia: f. 52-53.
1. Telefonia pela internet. 2. Sistemas telefônicos. 3. Redes de  
computação - Protocolos. I. Universidade Federal do Paraná. II.  
García Fernández, Evelio Martín. III. Título.

CDD: 004.6

# **TERMO DE APROVAÇÃO**

Franco Brandelero

## **Um Esquema de Alocação de Largura de Banda Dinâmica para Transmissão VoIP via Satélite**

Dissertação como requisito parcial para obtenção do grau de Mestre no Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Paraná.

---

Prof. Dr. Evelio Martín García Fernández – Orientador  
Universidade Federal do Paraná

---

Prof. Dr. Eduardo Parente Ribeiro – Convidado  
Universidade Federal do Paraná

---

Prof. Dr. Carlos Marcelo Pedroso – Convidado  
Universidade Federal do Paraná

---

Prof. Dr. Marcelo Eduardo Pellenz – Convidado  
Pontifícia Universidade Católica do Paraná

Curitiba, 20 de Agosto de 2010

## **RESUMO**

Hoje em dia é possível oferecer serviços de telefonia às regiões que se encontram distantes dos grandes centros ou que não dispõem de uma infraestrutura adequada, através de uma rede VoIP via satélite. A tecnologia VoIP permite um uso mais eficiente da largura de banda através de facilidades como compressão de áudio e supressão de silêncio. A supressão de silêncio, no entanto, apresenta uma incompatibilidade com o esquema padrão de alocação de banda dinâmica atualmente em uso nas redes satelitais.

Este trabalho apresenta um esquema de alocação de banda dinâmica, definido a partir de modelagem matemática, que além de visar um uso mais eficiente da banda prioriza a qualidade de voz. Através do simulador de redes de pacotes NS-2, o proposto esquema de alocação é comparado com o esquema padrão em vários cenários e a avaliação do desempenho se dá pela métrica MOS obtida através do Modelo-E.

Palavras-chave: DVB-RCS. VoIP. VAD. NS-2. MOS.

## **ABSTRACT**

Nowadays it is possible to offer telephony services to regions that are remote from major centers or who do not have an adequate infrastructure, through a VoIP network via satellite. VoIP technology allows more efficient use of bandwidth through facilities such as audio compression and silence suppression. The silence suppression, however, presents a mismatch with the standard scheme of dynamic bandwidth allocation currently used in satellite networks.

This paper presents a scheme of dynamic bandwidth allocation, defined based on mathematical modeling, which in addition to seeking a more efficient use of bandwidth prioritize voice quality. Through the packet network simulator NS-2, the proposed allocation scheme is compared with the standard scheme in various settings and performance evaluation is given by the MOS metric obtained from the E-Model.

Key-words: DVB-RCS. VoIP. VAD. NS-2. MOS.

## LISTA DE FIGURAS

FIGURA 1 - Requisição e alocação de banda.....	15
FIGURA 2 - Porção de fala e porção de silêncio em uma chamada VoIP .....	24
FIGURA 3 – Representação da transição entre os estados.....	25
FIGURA 4 - Topologia para medição de banda .....	32
FIGURA 5 - Probabilidade de perdas de pacotes em função da capacidade do enlace de saída .....	33
FIGURA 6 - Probabilidade de perdas de pacotes em função da probabilidade de bloqueio.....	34
FIGURA 7 - Comparação entre as taxas medidas e as taxas calculadas para alocação.....	35
FIGURA 8 - Comparação entre as taxas medidas e as taxas calculadas para alocação para poucas chamadas.....	36
FIGURA 9 – Topologia para simulação.....	37
FIGURA 10 - Transmissão de voz com esquema padrão de alocação.....	40
FIGURA 11 - Transmissão de voz e dados com esquema padrão de alocação .....	42
FIGURA 12 - Transmissão de voz com VAD usando esquema padrão de alocação	43
FIGURA 13 - Transmissão de voz com VAD e dados usando esquema padrão de alocação.....	44
FIGURA 14 - Transmissão de voz com VAD e dados usando pré-alocação de banda .....	45
FIGURA 15 - Transmissão de voz com VAD e dados usando alocação padrão em alto tráfego .....	47
FIGURA 16 - Transmissão de voz com VAD e dados usando pré-alocação em alto tráfego .....	48

## LISTA DE SIGLAS

ATM	- Asynchronous Transfer Mode
AVBDC	- Absolute Volume Based Dynamic Capacity
BoD	- Bandwidth on Demand
CF-DAMA	- Combined Free-Demand Assignment Multiple Access
CNG	- Comfort Noise Generator
CRA	- Constant Rate Assignment
DAMA	- Demand-Assignment Multiple Access
DBA	- Dinamic Bandwidth Allocation
DVB	- Digital Video Broadcasting
DVB-RCS	- DVB-Return Channel via Satellite
DVB-S	- DVB-Satellite
DVB-S2	- DVB-S version 2
FCA	- Free Capacity Assignment
fdp	- Função Densidade de Probabilidade
FDP	- Função Distribuição de Probabilidade
FTP	- File Transfer Protocol
HTTP	- Hypertext Transfer Protocol
IP	- Internet Protocol
ITU	- International Telecommunication Union
LAN	- Local Area Network
MAC	- Media Access Control
MF-TDMA	- Multi-Frequency-Time Division Multiple Access
MOS	- Mean Opinion Score
MPEG-2	- Motion Picture Expert Group
MPEG2-TS	- MPEG-2 Transport Stream
NCC	- Network Control Center
NS-2	- Network Simulator version 2
PGMU	- Plano Geral de Metas para a Universalização
QoS	- Quality of Service
RBDC	- Rate-based dynamic capacity
RCST	- Return Channel Satellite Terminal
RLR	- Receive Loudness Rating
RSVP	- ReSource reserVation Protocol
RTP	- Real Time Protocol
RTPC	- Rede de Telefonia Pública Comutada
RTT	- Rount Trip Time
SAC	- Satellite Access Control
SIP	- Session Initiation Protocol
SLR	- Send Loudness Rating



TBCP	- Terminal Burst Time Plan
TDM	- Time Division Multiplex
TDMA	- Time Division Multiple Access
UDP	- User Datagram Protocol
VAD	- Voice Activity Detection
VBDC	- Volume-based dynamic capacity
VoIP	- Voice over IP
VSAT	- Very Small Aperture Terminal

## LISTA DE SÍMBOLOS

$R$	- Escalar resultante do Modelo-E
$Ro$	- Relação sinal-ruído
$Is$	- Imparidades simultâneas ao sinal de voz
$Id$	- Imparidade causada pelo atraso
$Ieef$	- Imparidade causada pelo codec
$A$	- Fator de vantagem
$Ta$	- Retardo absoluto
$X$	- Variável auxiliar
$Ie$	- Variável dependente do codec
$Bpl$	- Variável dependente do codec
$P_{pl}$	- Probabilidade de perdas de pacotes
$\alpha$	- Inverso da duração média da porção de fala
$\beta$	- Inverso da duração média da porção de silêncio
$L$	- Inverso da duração média entre duas porções de fala
$q$	- Quantidade de chamadas no estado de fala
$n$	- Quantidade de chamadas simultâneas
$E_q$	- Estado do sistema quando há $q$ chamadas no estado de fala
$P_q$	- Proporção de tempo que o sistema permanece no estado $E_q$
$k$	- Variável auxiliar
$Vaf$	- Proporção de atividade de voz
$P(n, Vaf, q)$	- Probabilidade de bloqueio
$P_{max}$	- Maior valor aceitável da probabilidade de bloqueio
$B$	- Largura de banda do enlace de saída
$S$	- Tamanho do pacote
$T$	- Período de paquetização
$A(t)$	- FDP do tempo entre chegadas
$a(t)$	- fdp do tempo entre chegadas
$p$	- probabilidade de chegada dos pacotes
$N$	- Número médio de pacotes em uma porção de fala
$\lambda$	- Taxa média de chegadas de pacotes
$b(t)$	- fdp do tempo de serviço
$\mu$	- capacidade do enlace de saída em pacotes por segundo
$\delta(t)$	- Função impulso unitário
$w(t)$	- fdp do tempo de espera de um pacote em uma fila
$c(t)$	- fdp auxiliar
$J_B$	- Tamanho do buffer antijitter em unidades de tempo

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>11</b>
1.1 Contexto.....	11
1.2 Objetivo.....	11
1.3 Contribuições .....	12
1.4 Estrutura da Dissertação.....	12
<b>2 CONCEITOS FUNDAMENTAIS .....</b>	<b>14</b>
2.1 Padrão DVB-RCS .....	14
2.2 Implicações dos Diferentes Esquemas de Alocação de Largura de Banda .....	17
2.3 Visão Geral VoIP .....	18
2.4 Supressão de Silêncio .....	19
2.5 Supressão de Silêncio e Largura de Banda Dinâmica .....	19
2.6 Modelo-E.....	20
2.7 Estudos Anteriores.....	22
<b>3 TRÁFEGO VoIP COM SUPRESSÃO DE SILÊNCIO .....</b>	<b>24</b>
3.1 Modelagem da Fonte a Nível de Chamadas.....	24
3.2 Modelagem da Fonte a Nível de Pacote .....	27
3.3 Banda Alocada e Banda Ocupada.....	32
<b>4 SIMULAÇÃO DE DESEMPENHO USANDO NS-2 .....</b>	<b>37</b>
4.1 Cenário 1 – Poucas chamadas por VSAT.....	37
4.1.1 Simulação com Esquema Padrão de Alocação sem VAD.....	39
4.1.2 Simulação com Esquema Padrão de Alocação com VAD.....	42
4.1.3 Simulação com Pré-Alocação de Banda com VAD .....	45
4.1.4 Resumo dos resultados do Cenário 1 .....	46
4.2 Cenário 2 – Várias chamadas por VSAT .....	46
4.2.1 Simulação com Esquema Padrão de Alocação com VAD.....	47
4.2.2 Simulação com Pré-Alocação de Banda com VAD .....	47
4.2.3 Resumo dos resultados do Cenário 2 .....	49
<b>CONCLUSÃO .....</b>	<b>50</b>
<b>REFERÊNCIAS.....</b>	<b>52</b>
<b>APÊNDICE A – Equação Integral de Lindley.....</b>	<b>54</b>
<b>APÊNDICE B – Valores Padrão do Modelo-E.....</b>	<b>56</b>
<b>APÊNDICE C – SCRIPTS NS-2.....</b>	<b>57</b>
C.1 Voz sem VAD com Esquema Padrão .....	57
C.2 Voz sem VAD + Dados com Esquema Padrão .....	60
C.3 Voz com VAD e Esquema Padrão .....	64
C.4 Voz com VAD + Dados e Esquema Padrão.....	68
C.5 Voz com VAD + Dados e Pré-Alocação de Banda.....	73
C.6 Voz com VAD + Dados e Alocação Padrão em alto tráfego .....	81
C.7 Voz com VAD + Dados com Pré-Alocação em alto tráfego .....	85

# 1 INTRODUÇÃO

## 1.1 Contexto

Em 1998 foi aprovado no Brasil, através do decreto 2.592/1998 o primeiro Plano Geral de Metas para a Universalização (PGMU) e em 2003 através do decreto 4.769/2003 teve o segundo que entre outras, estabelece a seguinte meta: para localidades de 100 a 300 habitantes ter pelo menos um telefone público instalado em local acessível 24 horas por dia, com capacidade de originar e receber chamadas de longa distância nacional e internacional [1] [2].

A maneira mais rápida de se levar serviços de telefonia a essas localidades de baixa densidade que se encontram em regiões distantes é através de uma rede via satélite. Um dos padrões satelitais mais usados atualmente é o DVB-S (*Digital Video Broadcasting-Satellite*) que permite o uso da tecnologia de Voz sobre IP (VoIP) para telefonia.

Numa rede via satélite a largura de banda tem um custo muito alto, mas com o uso de VoIP há uma grande economia de banda em relação ao TDM (*Time Division Multiplex*) convencional devido à compressão dos *codecs* de voz e a supressão de silêncio. A supressão de silêncio, no entanto, ainda não é comumente usada numa rede DVB devido à incompatibilidade com o seu esquema padrão de alocação dinâmica da banda.

## 1.2 Objetivo

Este estudo tem como objetivo determinar um esquema de alocação dinâmica de banda que seja compatível com supressão de silêncio de modo a se ter um uso mais eficiente da largura de banda disponível, ou seja, fazer uso do benefício da supressão de silêncio – economia de largura de banda – e ao mesmo

tempo manter a qualidade de voz em um nível aceitável, inclusive em condições de baixo tráfego como o observado nas regiões atendidas pelo PGMU.

Partindo-se da teoria das filas será feita a modelagem do sistema (tanto a nível de chamadas como a nível de pacotes) para assegurar o embasamento teórico do esquema de alocação de banda dinâmica. Essa modelagem servirá de base para simulações computacionais de uma rede via satélite onde serão observados parâmetros como perda de pacotes, *jitter* e a latência fim-a-fim dos pacotes. Esses parâmetros são avaliados pelo Modelo-E [3], que fornece uma estimativa da qualidade percebida que pode ser mapeada na escala MOS (*Mean Opinion Score*) [4].

### 1.3 Contribuições

Trabalhos anteriores foram feitos visando-se um uso mais eficiente da banda satelital quando se tem supressão de silêncio, porém alguns métodos não levam em consideração a qualidade da chamada para o caso de poucas chamadas simultâneas e outros não fazem uso eficiente da banda. O presente trabalho visa priorizar a qualidade de voz em qualquer condição de tráfego e economizar banda em condições de tráfego mais intenso através de uma pré-alocação determinada estocasticamente.

### 1.4 Estrutura da Dissertação

Este trabalho está estruturado da seguinte forma. No segundo capítulo é feita uma breve descrição do padrão DVB-RCS (*DVB-Return Channel via Satellite*) e como este afeta a qualidade percebida de uma chamada VoIP quando há detecção de atividade de voz e são apresentados alguns estudos que visam melhorar essa qualidade. No terceiro capítulo é apresentada a modelagem que se propõe para prever o consumo de banda em chamadas VoIP com supressão de silêncio ativo. No quarto capítulo é aplicado o método de alocação desenvolvido em um sistema

DVB-RCS simulado e por fim no quinto capítulo uma discussão sobre os resultados e sugestões para trabalhos futuros são apresentadas.

## 2 CONCEITOS FUNDAMENTAIS

### 2.1 Padrão DVB-RCS

Iniciado em 1993, o consorcio internacional europeu DVB publicou em 1999, uma família de especificações de transmissão digital, baseada em técnicas de transmissão e compressão de vídeo MPEG-2 (*Motion Picture Expert Group*). Em cada especificação, os dados são transportados dentro de fluxos de transporte chamados MPEG2-TS (*MPEG-2 Transport Stream*). Adaptado para sistemas via satélite, o DVB-S define um dos formatos mais utilizados para TV digital.

Para apoiar os emergentes serviços de Internet via satélite, a sua segunda geração (DVB-S2) [5], com sua primeira versão publicada em março de 2000, inclui a transmissão de conteúdo multimídia e uma variedade de serviços *unicast* e *multicast*. Essa ampla possibilidade do sistema DVB-S2 necessita de interatividade, e assim, um enlace de retorno via satélite é mandatório. A atual contraparte do DVB-S2 para o enlace de retorno é o padrão DVB-RCS.

O sistema DVB-RCS realiza um esquema de acesso múltiplo por divisão multi-tempo-frequência (MF-TDMA – *Multi-Frequency-Time Division Multiple Access*) onde o canal de retorno é segmentado em porções de tempo e frequência (“superquadros”) divididos em janelas de tempo de duração e largura de banda fixa ou variável durante as quais o terminal (RCST – *Return Channel Satellite Terminal*) transmite pacotes MPEG2-TS ou células ATM (*Asynchronous Transfer Mode*).

Considere uma rede via satélite como mostrado na FIGURA 1. O Centro de Controle da Rede (NCC – *Network Control Center*), que é a entidade que coleta as demandas, executa o algoritmo para Alocação de Banda Dinâmica (DBA – *Dinamic Bandwidth Allocation*) e envia o resultado de alocação de volta ao RCST.

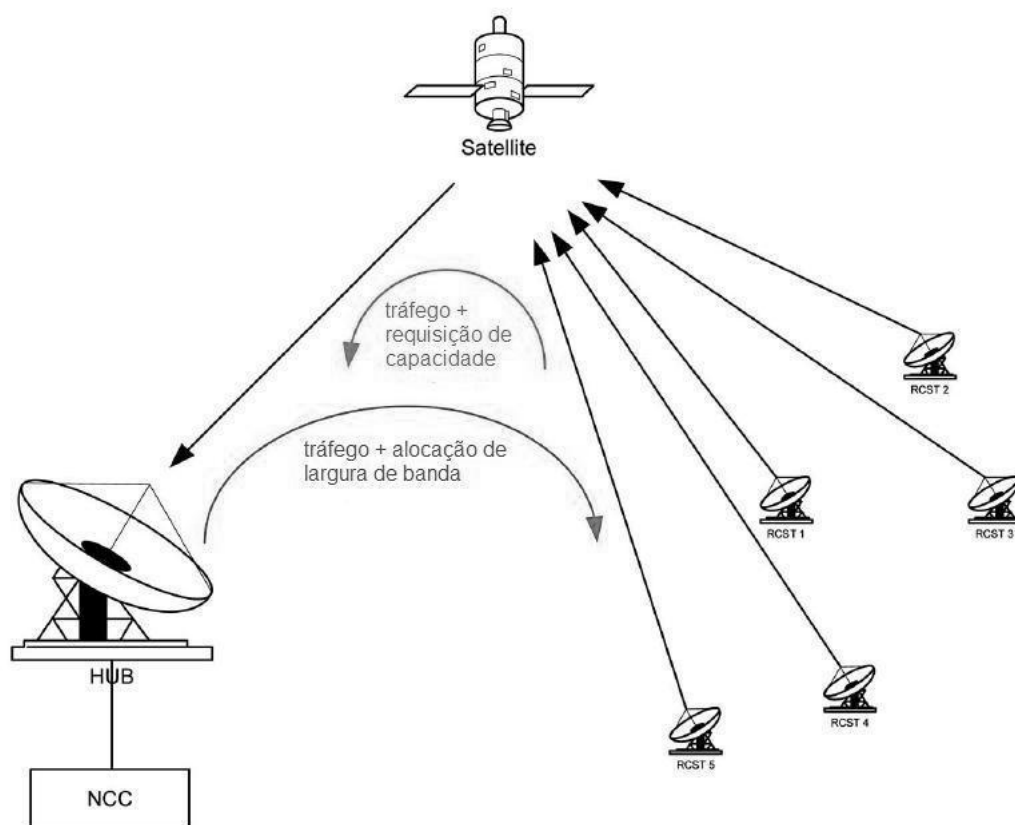


FIGURA 1 - Requisição e alocação de banda

As técnicas de alocação de banda permitidas pelo padrão DVB-RCS pertencem à classe chamada de reserva de recursos sob demanda ou acesso múltiplo por atribuição de demanda (DAMA – *Demand-Assignment Multiple Access*).

Mensagens para reserva de banda são enviadas pelos RCSTs para o NCC e um agendador centralizado aplica o algoritmo DBA. A alocação de banda é enviada de volta aos RCSTs no mínimo um tempo de “viagem ida e volta” (RTT – *Round Trip Time*) após o envio da requisição. As requisições de capacidade podem ser emitidas a cada RTT pelos RCSTs, entretanto, nem todas as estações irão requerer banda continuamente. É importante notar aqui o desafio do problema de alocação: enquanto o tráfego IP é por natureza sem conexão, os algoritmos DAMA estabelecem uma conexão sobre a interface aérea do DVB-RCS, a qual realiza acesso múltiplo por divisão multi-tempo-frequência (MF-TDMA).



As requisições geradas pelos RCSTs dependem do tráfego enfileirado nas filas da camada 2 de cada terminal e são enviadas usando o padrão de mensagens do controle de acesso via satélite (SAC – *Satellite Access Control*).

O padrão DVB-RCS define os seguintes cinco tipos de requisições de capacidade:

- Atribuição de taxa constante (CRA - *Constant Rate Assignment*): O RCST requisita uma taxa constante todo o tempo.
- Capacidade dinâmica baseada em taxa (RBDC - *Rate-based dynamic capacity*): Uma requisição de banda permanece ativa a menos que seja atualizada ou expirada. Em contraste ao CRA, a estratégia RBDC permite multiplexação estatística entre muitos RCSTs, resultando num uso mais eficiente da banda.
- Capacidade dinâmica baseada em volume (VBDC - *Volume-based dynamic capacity*): É requerida capacidade para uma determinada quantidade de volume para transmitir informações, sendo as requisições cumulativas, isto é, cada requisição é adicionada à requisição anterior do mesmo RCST. O total acumulado será reduzido pela quantidade atribuída em cada superquadro.
- Capacidade dinâmica baseada em volume absoluto (AVBDC - *Absolute Volume Based Dynamic Capacity*): É requerida capacidade para uma determinada quantidade de volume para transmitir informações, sendo cada requisição absoluta, isto é, substitui a requisição anterior.
- Atribuição de capacidade livre (FCA - *Free Capacity Assignment*): Uma capacidade de volume é atribuída ao RCST da capacidade total que de outra maneira não seria usada.

As requisições geradas por todos os terminais constituem entradas para o problema de alocação. Note-se que devido à latência do sistema, a predição de tráfego pode desempenhar um papel importante. Para cada atualização da alocação de banda, o NCC envia aos terminais o plano de tempo de rajadas do terminal (TBCP - *Terminal Burst Time Plan*). Nesse plano está a informação de quais janelas de tempo estão designadas a cada terminal. Com essa alocação, o terminal agenda o tráfego armazenado nas filas MAC.

O retardo de acesso – intervalo entre a chegada do pacote na fila do terminal e o tempo quando a capacidade alocada pode ser usada para transmitir o

pacote – é uma característica chave do esquema de largura de banda sob demanda (*Bandwidth on Demand* - BoD). Longos retardos têm um impacto negativo. O retardo de acesso varia de acordo com o intervalo de monitoração/alocação, estratégia de requisição de capacidade, estratégia de sinalização de requisição e esquema de alocação de largura de banda.

Para aplicações VoIP o esquema mais adequado é uma combinação CRA+RBDC [9], onde o CRA garante uma banda mínima para fins de sinalização e depois o RBDC é utilizado para a alocação conforme a demanda.

## 2.2 Implicações dos Diferentes Esquemas de Alocação de Largura de Banda

Numa rede satelital DVB-RCS os recursos de retorno são compartilhados entre os terminais. O compartilhamento é resolvido pelo uso de um protocolo da família CF-DAMA (*Combined Free-Demand Assignment Multiple Access*) para atribuição de capacidade. As categorias de capacidade formam a base dos diferentes esquemas de alocação de banda [6]:

**Atribuição de taxa fixa** provê os usuários com CRA estático. A capacidade fica alocada independente se o terminal tem ou não tráfego para transmitir.

**Atribuição de taxa fixa BoD** provê capacidade quando o terminal precisa dela (por exemplo, pela duração de uma sessão VoIP).

**Atribuição de taxa variável BoD** é um esquema dinâmico onde um número variável de janelas de tempo são atribuídas baseadas nas requisições. Pode ser baseada em RBDC ou VBDC.

O esquema de **atribuição livre** distribui a capacidade entre todos os terminais.

Esquemas de taxa fixa são otimizados para retardo e *jitter*, mas oferecem baixa escalabilidade. Um esquema de taxa fixa BoD também é livre de *jitter*, oferece melhor eficiência e boa responsividade. Atribuição de taxa variável BoD é otimizada para eficiência, o retardo de acesso pode ser baixo dependendo das estratégias de requisição e sinalização. E por fim, o esquema de atribuição livre é inapropriado para o serviço de VoIP.

## 2.3 Visão Geral VoIP

Dentro do contexto de transmissão via satélite, tem-se a seguir uma breve explicação dos parâmetros de performance relevantes a VoIP [6]:

**Retardo** (*Delay*) ou **Atraso fim-a-fim** é o tempo gasto pelo pacote entre a origem e o destino. Para uma rede com satélite ele é predominantemente o retardo de propagação, tipicamente 270 ms para satélites geoestacionários (240 ms no enlace satelital e o restante é devido aos enlaces terrestres e aos tempos de processamento nos nós intermediários). Apesar de esse valor ser alto ainda está abaixo do limite aceitável de 400 ms [7]. O tempo de ida-e-volta (RTT) é definido como o tempo entre a transmissão de um pacote até a sua resposta ser recebida.

**Jitter** é definido como variação do retardo [8]. O *jitter* pode levar distorções na voz devido a *buffer* curto ou aumento repentino no retardo. Para VoIP o *jitter* deve ser suavizado no receptor, ao custo de retardo adicional.

**Perda de pacotes** é o resultado do retardo de propagação e imparidades no canal (ruído, interferência). Ela pode causar lacunas na sequência de áudio. Algoritmos no receptor podem ajudar a cancelar perdas individuais mas geralmente não para perdas em rajadas, mais comuns em enlaces aéreos do que em redes cabeadas.

**Largura de banda.** A banda requerida para voz comprimida depende do codec de voz. A eficiência da banda é associada com a complexidade, algoritmo e retardo de processamento.

Os três primeiros parâmetros acima são parâmetros de qualidade de serviço (*Quality of Service* – QoS), enquanto o último está associado com utilização de recursos de rede.

Durante o processo de codificação a voz é digitalizada e segmentada em blocos tipicamente de 10-20 ms. Um codificador de voz comprime os blocos (quadros) de voz em um conjunto de parâmetros e num sinal de excitação que pode ser mais eficientemente representado (menos bits) do que a voz original digitalizada.

G.711, G.723, e G.729 são recomendações ITU (*International Telecommunication Union*) amplamente usadas para codificação de áudio. G.711 define o formato PCM (*Pulse Code Modulation*) de 64 kbps que provê apenas compressão logarítmica em amplitude. Em contraste, G.729 e G.723 comprimem a voz para 8 kbps ou 6,3 (5,3) kbps.

Pacotes VoIP são pequenos, com carga de 20-150 bytes e cabeçalhos RTP/UDP/IP de 40 bytes quando não comprimidos. Por exemplo, um fluxo G.729 a 8 kbps com 4 quadros de 10 ms por pacote IP, irá requerer uma capacidade de 16 kbps no nível IP.

## 2.4 Supressão de Silêncio

Numa conversação normal entre duas pessoas, cada transmissor está ativo por volta de 40% do tempo [6], o resto é silêncio. Durante os períodos de silêncio o *codec* codifica apenas ruído de fundo. Usando detecção de atividade de voz (*Voice Activity Detection – VAD*), é possível detectar os quadros de silêncio e suprimir a codificação; a fonte se torna inconstante (LIGADO/DESLIGADO). Para prevenir o desconfortável silêncio morto, um gerador de ruído de conforto (*Comfort Noise Generator – CNG*) pode ser ativado.

## 2.5 Supressão de Silêncio e Largura de Banda Dinâmica

Apesar da aplicação de VAD oferecer uma solução para a eficiência da largura de banda, combinada com um padrão de mecanismo de requisição RBDC pode levar a retardos adicionais. Essa é uma das razões para a degradação de desempenho, junto com o fato de que os sistemas satelitais são dimensionados assimetricamente (tipicamente 8:1 ou 4:1), oferecendo menos largura de banda ao canal de retorno [9].

Se um esquema RBDC nominal é usado, não é possível manter uma taxa contínua, pois a capacidade não é requisitada durante o período “desligado”. Isso

leva a uma re-ocorrência adicional de retardo de 0,4 – 1,5 segundos cada vez que a capacidade é re-requisitada para um período “ligado”. Quando o VAD está habilitado, a qualidade percebida pelo usuário se reduz significativamente, pois o algoritmo de controle não consegue sincronizar com a taxa codificada quando a aplicação oscila entre silêncio e fala.

Tráfego com taxa variável, como é o caso do VoIP com VAD, pode ser usado em conjunto com técnicas de alocação de capacidade (BoD), porém esta deve ser adaptada para as especificações de tráfego VoIP para que a qualidade da voz seja mantida em um nível aceitável.

## 2.6 Modelo-E

Tradicionalmente tem-se usado a escala MOS (*Mean Opinion Score*) [4] para medir a percepção subjetiva de uma comunicação de voz. A escala MOS vai de 1 a 5 sendo 1 péssimo e 5 excelente e o resultado final é a média aritmética das opiniões de todos os entrevistados.

Como esse método é muito caro e demorado, a ITU-T propôs a recomendação G.107 onde é definido o algoritmo Modelo-E [3], que tem se mostrado uma ferramenta útil para planejamento de transmissão, ao considerar os efeitos combinados de variações de vários parâmetros que afetam a qualidade de uma conversa telefônica.

O resultado do modelo-E é um fator escalar que é composto dos seguintes elementos:

$$R = Ro - Is - Id - Ieef + A \quad (2.1)$$

$Ro$  representa a relação sinal-ruído. O fator  $Is$  é uma combinação de todas as imparidades que ocorrem mais ou menos simultaneamente com o sinal de voz. O fator  $Id$  representa a imparidade causada pelo atraso e o fator  $Ieef$  representa imparidades efetivas causadas por codecs de baixa taxa de bits. O fator vantagem  $A$  permite a compensação dos fatores de imparidade quando há outras vantagens de acesso para o usuário.

O mapeamento entre o fator  $R$  e a escala MOS é dado pela seguinte fórmula:

$$\text{Para } R < 0: \quad MOS = 1$$

$$\text{Para } 0 < R < 100: \quad MOS = 1 + 0,035R + R(R - 60)(100 - R)7 \cdot 10^{-6} \quad (2.2)$$

$$\text{Para } R > 100: \quad MOS = 4,5$$

A relação entre a satisfação do usuário e o fator  $R$  é dado pelo seguinte quadro:

<b><math>R</math> (limite inferior)</b>	<b>MOS (limite inferior)</b>	<b>Satisfação do usuário</b>
90	4,34	Muito satisfeitos
80	4,03	Satisfeitos
70	3,60	Alguns usuários insatisfeitos
60	3,10	Muitos usuários insatisfeitos
50	2,58	Praticamente todos usuários insatisfeitos

QUADRO 1 – Relação entre o fator  $R$  e a satisfação do usuário

Usando os valores padrão para os parâmetros que não variam durante o planejamento (Apêndice B), o cálculo resulta numa alta qualidade com fator  $R=93,2$ .

Considerando apenas as imparidades de rede (retardo e perda de pacotes) o fator  $R$  pode ser reescrito como [10]:

$$R = 93,2 - Id - I_{eef} + A \quad (2.3)$$

Assumindo perfeito cancelamento de eco, o fator  $Id$  representa a imparidade causada pelo retardo absoluto  $Ta$ . O retardo absoluto será a soma do período de paquetização, o tempo de propagação fim-a-fim e o retardo adicionado pelo *buffer antijitter*.

Para  $Ta < 100$  ms:

$$Id = 0 \quad (2.4)$$

Para  $Ta > 100$  ms:

$$Id = 25 \left\{ \left( 1 + X^6 \right)^{\frac{1}{6}} - 3 \left( 1 + \left[ \frac{X}{3} \right]^6 \right)^{\frac{1}{6}} + 2 \right\} \quad (2.5)$$

Onde

$$X = \frac{\lg\left(\frac{Ta}{100}\right)}{\lg 2} \quad (2.6)$$

O fator  $I_{eef}$  é calculado da seguinte maneira:

$$I_{eef} = I_e + (95 - I_e) \cdot \frac{P_{pl}}{P_{pl} + B_{pl}} \quad (2.7)$$

Onde  $I_e$  e  $B_{pl}$  são valores tabelados dependentes do codec usado e  $P_{pl}$  é a probabilidade de ocorrência de perdas de pacotes. Pacotes que chegam ao receptor com um retardo maior que  $Ta$  (causado pelo *jitter*) devem ser contabilizados como perdidos.

O fator vantagem  $A$  pode assumir valores conforme quadro abaixo.

Exemplo de sistema de comunicação	Valor máximo de A
Convencional (com fio)	0
Mobilidade em um prédio	5
Mobilidade em uma área geográfica	10
Acesso a locais de difícil acesso, conexão via satélite	20

QUADRO 2 – Relação entre sistema de comunicação e o fator A

## 2.7 Estudos Anteriores

Para resolver o problema da baixa qualidade de voz quando se tem VAD numa rede via satélite, foram propostos alguns métodos descritos a seguir.

A solução apresentada em [11] consiste em uma combinação de RBDC e VBDC. Neste caso o papel do VBDC é apoiar o tráfego de melhor esforço, se existe, e manter o RBDC continuamente enviando requisições de capacidade. O problema desse método é que para que ocorra o alívio de transição entre silêncio e fala é necessário uma grande quantidade de chamadas simultâneas (mais de vinte) no terminal VSAT, o que torna esse método inviável em localidades de baixa densidade de tráfego.

Em [9] é proposto uma alteração no algoritmo de requisição de banda que se baseia em estimar a taxa média ocupada pelas chamadas através da equação (2.8), onde  $S_F$  é o tamanho do quadro e a média do tempo entre chegadas de pacotes  $E(X_n)$  dado pela equação (2.9) leva em consideração a quantidade de chamadas simultâneas  $n$ , o período de paquetização  $T$ , a duração média do período de silêncio  $\beta^{-1}$ , e a probabilidade de chegada dos pacotes  $p$ .

$$\bar{R}_{VoIP} = S_F / E(X_n) \quad (2.8)$$

$$E(X_n) = \frac{\beta T + 1 - p}{n\beta} \quad (2.9)$$

O problema é que nem sempre a banda requerida corresponde à taxa média, principalmente em condições de baixo tráfego onde a média é muito inferior à banda necessária.

Outra solução também aparece em [12] onde é proposto um SIP Proxy com funcionalidades estendidas para automaticamente fazer a reserva de recursos para uma sessão multimídia. As funcionalidades que foram adicionadas ao SIP Proxy padrão são:

- Um analisador SDP de maneira que o Proxy seja capaz de entender as descrições de sessão definidas em formato SDP inclusas nas mensagens SIP.
- Uma tabela de mídia que é atualizada a cada estabelecimento de sessão SIP.
- Um mapeamento SDP/DiffServ de tipo de mídia para um serviço DiffServ.
- Um módulo QoS que faz a reserva de recursos correspondente à mídia na sessão SIP.

A desvantagem desse método é que a banda alocada corresponde a uma sessão sem VAD, o que realmente melhora a qualidade percebida, porém não faz um uso eficiente da largura de banda satelital.

Para resolver o problema de manter a qualidade da voz dentro de níveis aceitáveis e economizar largura de banda (quando possível), uma pré-alocação necessita ser feita no início da chamada. A quantidade de banda que deve ser alocada será determinada no próximo capítulo.



### 3 TRÁFEGO VoIP COM SUPRESSÃO DE SILÊNCIO

Quando há várias chamadas simultâneas a probabilidade de que todas as chamadas estejam no estado de FALA é pequena, então uma alocação estatística é mais apropriada.

Existem vários modelos que visam representar fontes de tráfego VoIP com supressão de silêncio, como os citados em [13], porém no cenário do PGMU o usuário se encontra atrás de um conversor analógico/VoIP, comumente chamado de *Media Gateway*, e é tarifado de acordo com as regras da RTPC (Rede de Telefonia Pública Comutada), portanto seu perfil de tráfego é o mesmo do observado naquela rede.

#### 3.1 Modelagem da Fonte a Nível de Chamadas

Um tráfego VoIP com detecção de atividade de voz apresenta um perfil LIGADO/DESLIGADO, os tempos LIGADOS e DESLIGADOS têm duração aleatória com distribuição exponencial [14] cujas médias são  $1/\alpha$  e  $1/\beta$  respectivamente, e sendo a média do período entre duas porções de fala  $1/L$  conforme representado na FIGURA 2.

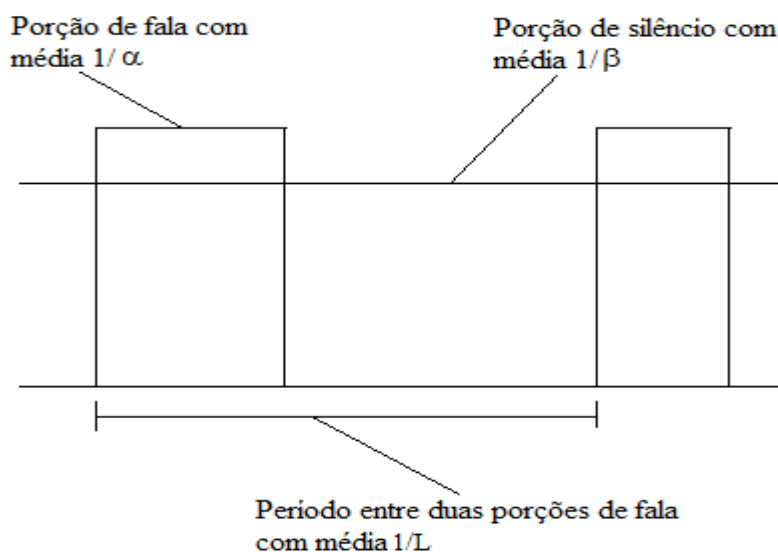


FIGURA 2 - Porção de fala e porção de silêncio em uma chamada VoIP

Usando a teoria das filas pode-se deduzir uma fórmula que represente a proporção em que todas as chamadas estão no estado de FALA. A dedução que aparece a seguir até a equação (3.6) é a mesma que pode ser encontrada em [15].

Quando o número de chamadas no estado de fala é  $q$ , de um total de  $n$  chamadas em andamento, o sistema está no estado  $E_q$  ( $q=0,1,\dots,n$ ). Seja  $P_q$  a proporção de tempo que o sistema permanece no estado  $E_q$ . Considerando primeiro o caso  $q < n$ . Seja  $L$  a taxa de mudança para o estado de fala para uma fonte, então a taxa na qual a transição  $E_q \rightarrow E_{q+1}$  ocorre é  $nLP_q$ . Agora considerando o caso  $q=n$ . Uma vez que o estado  $E_{n+1}$  representa um estado impossível, a taxa de transição  $E_n \rightarrow E_{n+1}$  é zero. Assim a taxa na qual a transição  $E_q \rightarrow E_{q+1}$  ocorre é  $nLP_q$  quando  $q=0,1,\dots,n-1$  e zero quando  $q=n$ .

Considerando-se agora a transição

$$E_{q+1} \rightarrow E_q \quad (q=0,1,\dots,n-1) \quad (3.1)$$

Sendo a duração média da porção de fala  $1/\alpha$ , então se há uma chamada e esta está no estado de fala, a média de mudanças para o estado de silêncio é  $\alpha$ . Se há duas chamadas e ambas no estado de fala, a média de mudanças para o estado de silêncio é  $2\alpha$  e assim sucessivamente. Conclui-se então que a taxa de mudanças para o estado de silêncio para  $q+1$  chamadas no estado de fala é  $(q+1)\alpha$ . Uma vez que o sistema está no estado  $E_{q+1}$  durante uma proporção de tempo  $P_{q+1}$ , então a transição  $E_{q+1} \rightarrow E_q$  ocorre a uma taxa  $(q+1)\alpha P_{q+1}$  (ver FIGURA 3).

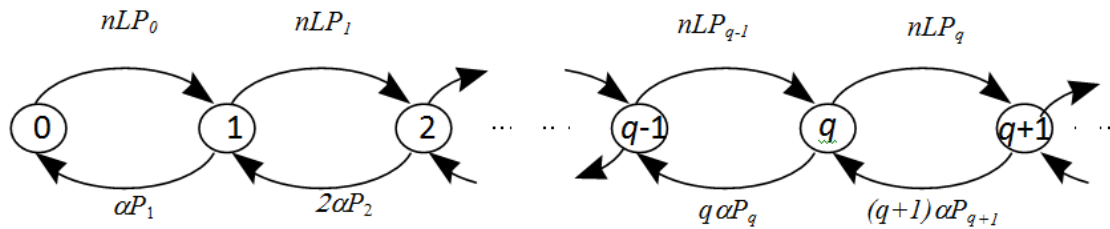


FIGURA 3 – Representação da transição entre os estados

Aplicando-se agora o princípio da conservação dos fluxos: Igualase, para cada valor do índice  $q$ , a taxa de ocorrências das transições  $E_q \rightarrow E_{q+1}$  com a taxa de ocorrências das transições  $E_{q+1} \rightarrow E_q$ . Assim tem-se a equação do equilíbrio estático ou da conservação dos fluxos:

$$nLP_q = (q+1)\alpha P_{q+1} \quad (q=0,1,\dots,n-1). \quad (3.2)$$

O resultado que expressa cada  $P_q$  em termos de  $P_0$  é

$$P_q = \frac{(nL/\alpha)^q}{q!} P_0 \quad (q=0,1,\dots,n). \quad (3.3)$$

Uma vez que os números  $P_q$  são proporções, sua soma deve ser unitária:

$$\sum_{q=0}^n P_q = 1 \quad (3.4)$$

Logo, podemos determinar  $P_0$ :

$$P_0 = \left( \sum_{k=0}^n \frac{(nL/\alpha)^k}{k!} \right)^{-1} \quad (3.5)$$

Assim obtemos para a proporção de tempo  $P_q$  em que  $q$  chamadas estão no estado de fala a equação [15]:

$$P_q = \frac{(nVaf)^q / q!}{\sum_{k=0}^n (nVaf)^k / k!} \quad (q=0,1,\dots,n). \quad (3.6)$$

Onde  $Vaf = (L/\alpha)$  representa a proporção de atividade de voz.

Se houver  $q$  canais alocados, pode-se reescrever essa equação, que também representa a probabilidade de bloqueio, da seguinte maneira:

$$P(n,Vaf,q) = \begin{cases} 1, & q = 0 \\ \frac{(nVaf)^q / q!}{\sum_{k=0}^q (nVaf)^k / k!}, & 0 < q < n \\ 0, & q = n \end{cases} \quad (3.7)$$

Fixando-se o valor máximo que  $P(n, Vaf, q)$  pode assumir, pode-se determinar o menor valor de  $q$  que atende  $n$  chamadas simultâneas. Uma vez determinado  $q$ , basta multiplicar esse valor pela taxa requerida para uma chamada VoIP e assim obtém-se a largura de banda necessária:

$$B = \frac{q \cdot S}{T} \quad (3.8)$$

Onde  $S$  é o tamanho do pacote e  $T$  é o período de paquetização do codec de voz.

Apesar de numa rede IP não existir o conceito de canais, a largura de banda ocupada por chamadas VoIP é sempre múltiplos de  $S/T$ . Também não existe um bloqueio de chamadas como ocorre nas redes TDM, porém cada vez que a necessidade de banda supera a banda alocada/disponível, há uma degradação na qualidade de todas as chamadas em curso.

Como determinar o maior valor aceitável de  $P(n, Vaf, q)$  é explicado na próxima seção.

### 3.2 Modelagem da Fonte a Nível de Pacote

Para chamadas VoIP com supressão de silêncio ativo, a função distribuição de probabilidade (FDP) do tempo entre chegadas para a superposição de  $n$  chamadas simultâneas [14] é:

$$A(t) = \begin{cases} 1 - (1 - \lambda t)^{n-1}, & 0 \leq t \leq T \\ 1 - \frac{(1 - p)^n e^{-\beta n(t-T)}}{(\beta T + 1 - p)^{n-1}}, & t \geq T \end{cases} \quad (3.9)$$

Onde

$p = \frac{N-1}{N}$  é a probabilidade de chegada dos pacotes;

$N = \frac{\alpha^{-1}}{T}$  é o número médio de pacotes em uma porção de fala;

$\lambda = \frac{Vaf}{T}$  é a taxa média de chegadas para uma chamada em pacotes por segundo;

$\beta^{-1}$  é a duração média do período de silêncio;

A sua função densidade de probabilidade (fdp) pode ser encontrada pela sua derivada:

$$a(t) = \frac{dA(t)}{dt} \quad (3.10)$$

O tempo de serviço de um pacote ao tempo corresponde ao tempo em que o pacote sai da fila até ser totalmente transmitido, ou seja, é a razão entre o tamanho do pacote e a capacidade do enlace de saída. Como para a maioria dos codecs VoIP todos os pacotes têm o mesmo tamanho durante a chamada (por exemplo a família G.7xx padronizada pela ITU-T), a fdp do tempo de serviço é dada por:

$$b(t) = \delta\left(t - \frac{1}{\mu}\right) \quad (3.11)$$

Onde  $\mu$  é a capacidade do enlace de saída em pacotes por segundo e  $\delta(t)$  é a função impulso unitário.

A fdp do tempo de espera  $w(t)$  do  $(j+1)$ -ésimo pacote que chega no RCST pode ser encontrada resolvendo-se a equação integral de Lindley (ver Apêndice A) [16]:

$$w_{j+1}(t) = \begin{cases} w_j(t) * c(t), & t > 0 \\ \delta(t) \int_{-\infty}^{0^+} [w_j(t) * c(t)] dt, & t = 0 \\ 0, & t < 0 \end{cases} \quad (3.12)$$

sendo

$$c(t) = a(-t) * b(t) \quad (3.13)$$

$$w_1(t) = \delta(t) \quad (3.14)$$

onde o símbolo  $*$  denota convolução.

Quando a fila atinge o equilíbrio estático temos:

$$w_{j+1}(t) = w_j(t) = w(t) \quad (3.15)$$

Para se estimar o nível de qualidade MOS através do modelo-E é necessário estimar a porcentagem de perdas para aplicá-lo na equação (2.7). Considerando-se que não haja perda de pacotes ocasionada pela rede e que o RCST é o único ponto de gargalo, os pacotes que têm um tempo de espera superior ao tempo do buffer *antijitter* são descartados no receptor. A probabilidade de perdas pode ser estimada pela equação a seguir:

$$P_{pl} = \int_{J_B}^{\infty} w(t)dt = 1 - \int_0^{J_B} w(t)dt \quad (3.16)$$

Onde  $J_B$  é o tamanho do buffer *antijitter* em unidades de tempo.

Para as funções de tempo entre chegadas e de tempo de serviço consideradas não existe equações fechadas para  $w(t)$  e  $P_{pl}$ , sendo portanto encontradas apenas por métodos numéricos.

Para se atingir o maior valor possível do MOS,  $P_{pl}$  deve ser zero. Para um número fixo de chamadas,  $P_{pl}$  deve ser calculado alterando-se o valor de  $\mu$  dentro dos seguintes limites:

$$n \cdot \lambda < \mu \leq \frac{n}{T} \quad (3.17)$$

O limite inferior vem do limite de convergência imposto pela teoria de filas, isto é, abaixo deste valor o tamanho da fila e o tempo de espera vão para o infinito [15]. É importante notar que esse valor também corresponde à taxa média para  $n$  chamadas.

O limite superior é a taxa usada quando o VAD está desativado e acima desse valor o tempo de espera é sempre zero.

Uma vez encontrado o valor de  $\mu$  que torna  $P_{pl} = 0$ , calcula-se seu equivalente em quantidade de canais:

$$q = \lceil \mu \cdot T \rceil \quad (3.18)$$

Onde o símbolo  $\lceil \rceil$  representa arredondamento para cima.

Aplica-se o valor de  $q$  na equação (3.7) para encontrar o maior valor aceitável da probabilidade de bloqueio.

Apesar de o valor de  $\mu$  já representar a largura de banda que deve ser alocada, seu cálculo é extremamente oneroso computacionalmente, o que o inviabiliza para uma aplicação em tempo real e aqui entra a necessidade da equação (3.7) que é muito mais simples e pode ser calculada em tempo real. Portanto o cálculo para alocação de banda deve ser feito em duas fases: Dimensionamento e Operação.

A fase de Dimensionamento é realizada apenas uma vez e pode ser dividida nos seguintes passos:

- 1 – Fixar o valor de  $n$  como sendo o número máximo de chamadas que irão trafegar pelo RCST;
- 2 – Calcular  $a(t)$  pela equação (3.10);
- 3 – Configurar o valor inicial de  $\mu$  como  $n\lambda$ ;
- 4 – Calcular  $b(t)$  pela equação (3.11);
- 5 – Calcular  $w(t)$  pela equação (3.15);
- 6 – Calcular  $P_{pl}$  pela equação (3.16);
- 7 – Se  $P_{pl} > 0$ , incrementar  $\mu$  e voltar ao passo 4; Se não calcular  $q$  pela equação (3.18);
- 8 – Calcular  $P(n, Vaf, q)$  pela equação (3.7). Esse é o maior valor aceitável da probabilidade de bloqueio  $P_{max}$ .

Na fase de Operação o cálculo é realizado em tempo real pelo RCST a cada início ou fim de uma chamada e pode ser dividida nos seguintes passos:

- 1 – Configurar valor inicial de  $q$  igual a 1;
- 2 – Para cada início ou fim de uma chamada, atualizar o valor de  $n$ ;
- 3 – Calcular  $P(n, Vaf, q)$  pela equação (3.7);
- 4 – Se  $P(n, Vaf, q) > P_{max}$ , incrementar  $q$  e voltar ao passo 3; Se não calcular  $B$  pela equação (3.8).

A sequência apresentada aqui para a fase de Operação é apenas didática, o cálculo de  $B$  pode ser otimizado, por exemplo, com o cômputo de  $B$  para vários valores de  $n$  durante a inicialização/configuração do RCST e os valores armazenados em memória, bastando apenas ser acessados a cada atualização de  $n$ .

O grande diferencial da metodologia de cálculo ora sendo apresentada é que a alocação é feita pela menor taxa que garante a maior qualidade e não pela média como descrito em [9].

A implementação deste algoritmo nos terminais RCSTs depende de uma interação entre as camadas de enlace e de aplicação para a troca de informação entre as camadas cada vez que se inicia uma nova chamada. Essa interação poderia ser feita, por exemplo, com a expansão da capacidade do terminal em analisar os pacotes, reconhecer os que são de sinalização VoIP, como SIP (*Session Initiation Protocol*) [17] ou Megaco [18], atualizando assim o número de chamadas ativas. Outra possibilidade seria a utilização de um protocolo de reserva de recursos, como o RSVP (*ReSource reserVation Protocol*) [19], entre o terminal RCST e o equipamento originador de tráfego VoIP. O método pelo qual a informação é passada ao algoritmo está além do escopo deste trabalho, e o restante da dissertação trata dos efeitos dessa alocação no tráfego de voz, independente de como é feita interação entre as camadas.



### 3.3 Banda Alocada e Banda Ocupada

Para verificar a confiabilidade da fórmula encontrada realizaram-se 100 simulações em um enlace terrestre na topologia da FIGURA 4 usando o NS-2 (*Network Simulator 2*), cada simulação com um número diferente de chamadas simultâneas. O atraso de propagação desse enlace é de 10 ms, porém qualquer valor pode ser usado aqui, pois o objetivo aqui não é avaliar a qualidade percebida mas sim qual a banda ocupada pelo tráfego RTP que usa UDP como camada de transporte, que é insensível ao atraso de propagação, diferentemente do TCP por exemplo.

Para simular o codec G.729annexB com VAD ativado [20] foi configurado um tráfego LIGADO/DESLIGADO onde os tempos LIGADO/DESLIGADO tem duração aleatória com distribuição exponencial, ou seja, taxa constante durante o tempo LIGADO e nenhum tráfego durante o período DESLIGADO. O período de packetização utilizado foi de 40 ms, resultando em pacotes de 94 bytes (14 bytes de cabeçalho Ethernet, 20 bytes de IP, 8 bytes de UDP, 12 bytes de RTP e 40 bytes de Payload) e numa taxa de 18,8 kbps no período de FALA, médias de 360 ms e 650 ms para os períodos de FALA e SILÊNCIO respectivamente, resultando num fator de atividade de voz de 35,6%, conforme estudos de [9] e [14], e com um *buffer antijitter* no receptor de 40 ms. O tempo de simulação é de 60 segundos. O enlace terrestre possui 2 Mbps, o que é mais que suficiente para trafegar 100 chamadas simultâneas sem VAD, garantindo assim que com VAD ativo nunca ocorra congestionamento.

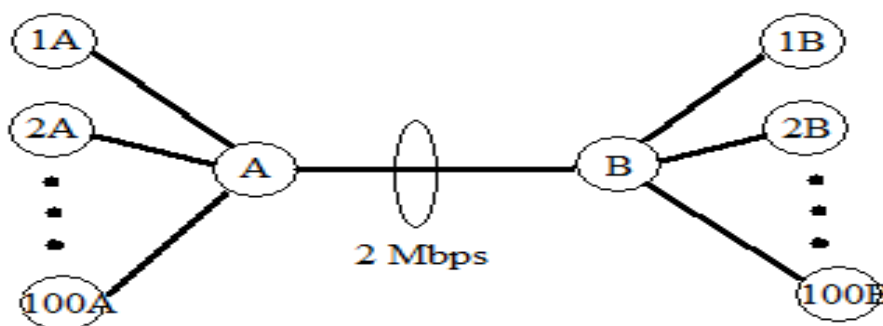


FIGURA 4 - Topologia para medição de banda

A probabilidade de perdas de pacotes para 100 chamadas simultâneas, calculada pela equação (3.16), variando-se  $\mu$  desde 891 pacotes por segundo (670 kbps) até obter-se uma probabilidade de perdas igual a zero em 1100 pacotes por segundo (1,88 Mbps), pode ser visto na FIGURA 5.

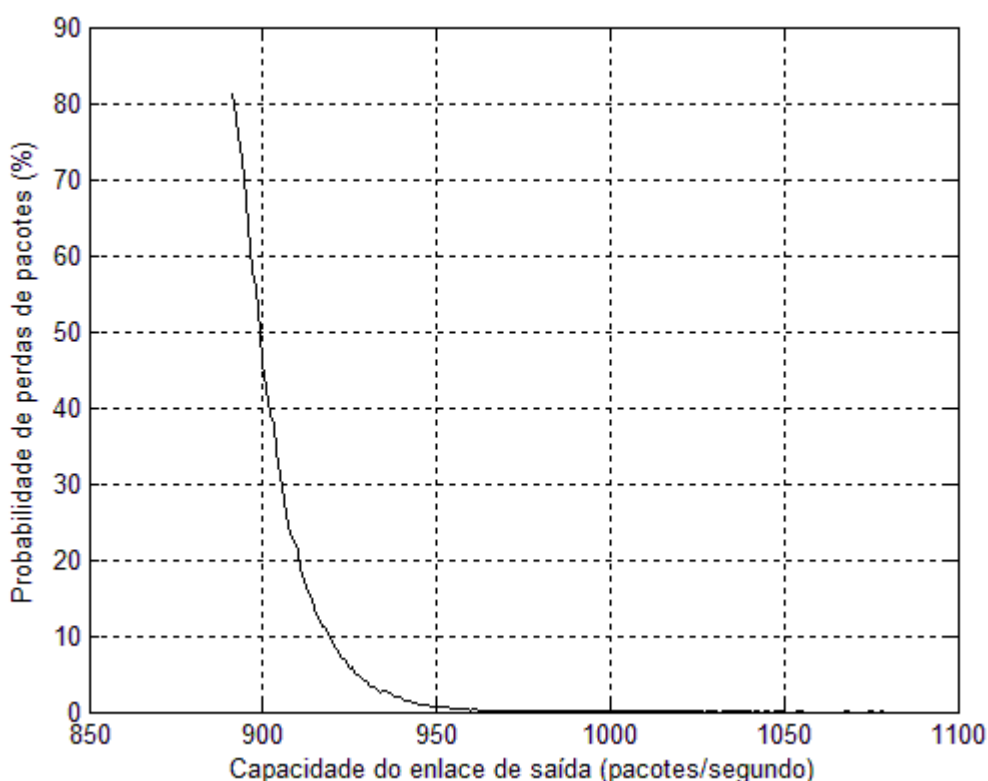


FIGURA 5 - Probabilidade de perdas de pacotes em função da capacidade do enlace de saída

Aplicando-se a equação (3.18) apenas para esses valores de  $\mu$ , e usando-se esse equivalente em quantidade de canais na equação (3.7), pode-se visualizar a relação entre probabilidade de perdas e probabilidade de bloqueio para esse cenário (FIGURA 6).

Como esperado, valores menores da probabilidade de bloqueio possuem uma menor probabilidade de perdas de pacotes. Durante o desenvolvimento dessa dissertação foi observado que as perdas de pacotes deixam de ser significativas a partir de 0,00001% portanto uma probabilidade de bloqueio de 3% é aceitável para

essa configuração e esse valor será usado para calcular a banda a ser alocada pela equação (3.8).

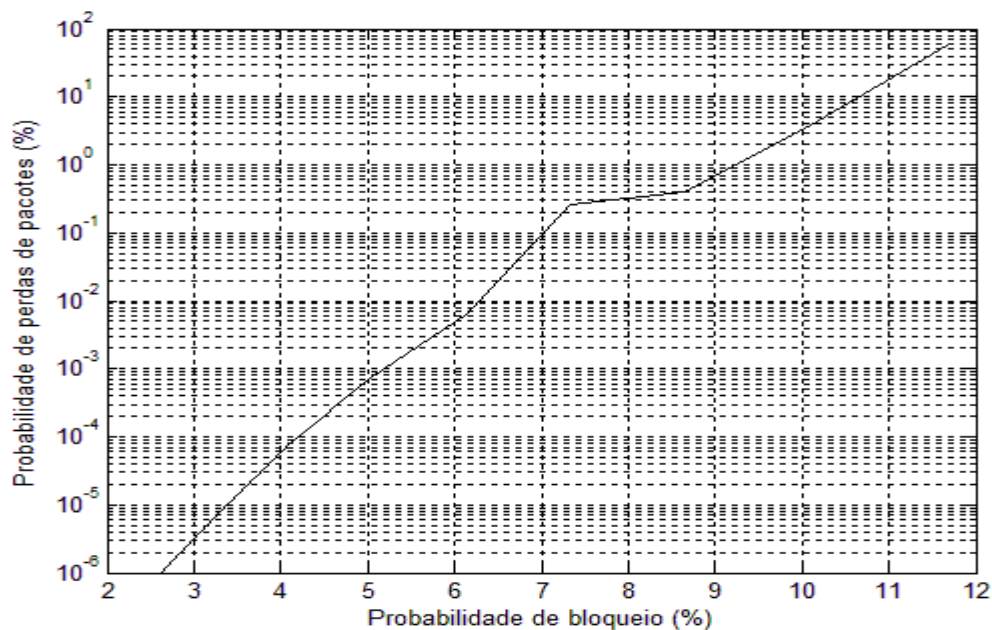


FIGURA 6 - Probabilidade de perdas de pacotes em função da probabilidade de bloqueio

Para cada simulação foi coletada a largura de banda máxima ocupada e comparada com a banda calculada com a equação (3.8). O resultado pode ser visto na FIGURA 7.

Para efeitos de comparação, a linha pontilhada corresponde à largura de banda que seria necessária quando o VAD está desabilitado, ou seja, é o limite superior da equação (3.17). A média teórica com VAD apresentada no gráfico é o limite inferior da equação (3.17), ou seja, é simplesmente a multiplicação do valor sem VAD pelo fator de atividade de voz, que é a conta normalmente feita pelos projetistas de rede VoIP para estimativa de largura de banda necessária, porém pode-se ver claramente que esse valor está sempre abaixo da banda máxima medida, resultando numa degradação da qualidade de voz.

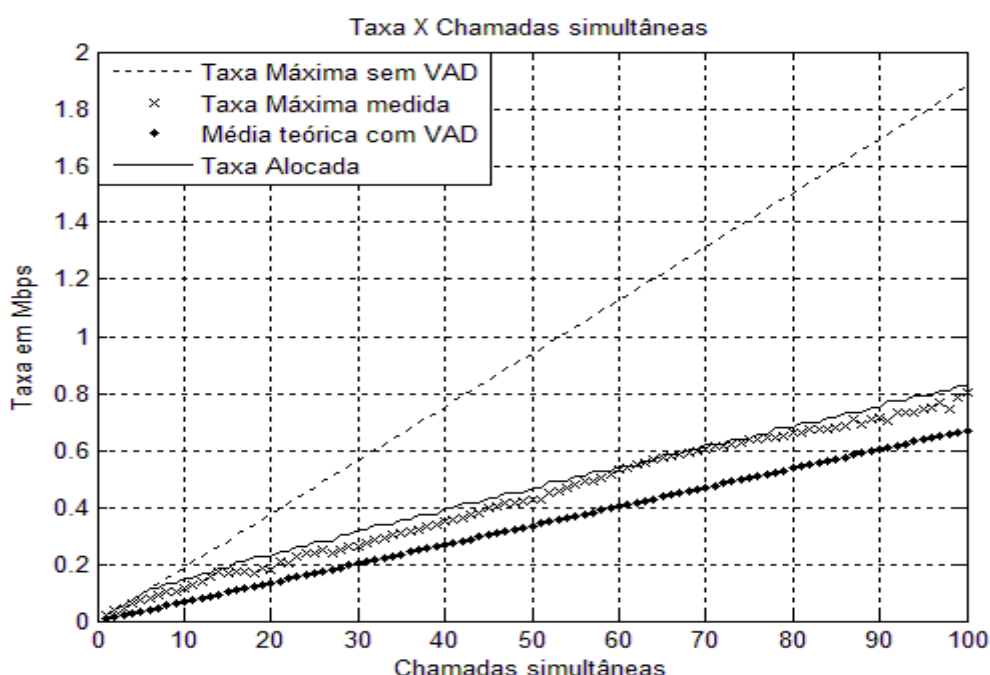


FIGURA 7 - Comparação entre as taxas medidas e as taxas calculadas para alocação

A taxa calculada com a equação (3.8), que representa a quantidade de banda que deve ser alocada, é sempre maior que a banda máxima ocupada, o que significa que esse método pode ser usado para alocação de banda quando se tem o VAD habilitado. A diferença entre a taxa máxima sem VAD e a taxa alocada representa o ganho de largura de banda, ou seja, quanto de banda está sendo economizado.

Na FIGURA 8, que é uma ampliação da figura anterior, pode-se observar o caso de baixo tráfego, quando há até 6 chamadas simultâneas em cada RCST, onde aparentemente não há nenhuma vantagem, pois a banda alocada é a mesma banda que seria ocupada caso o VAD estivesse desabilitado, porém essa é a única solução que mantém a qualidade de voz em um nível aceitável.

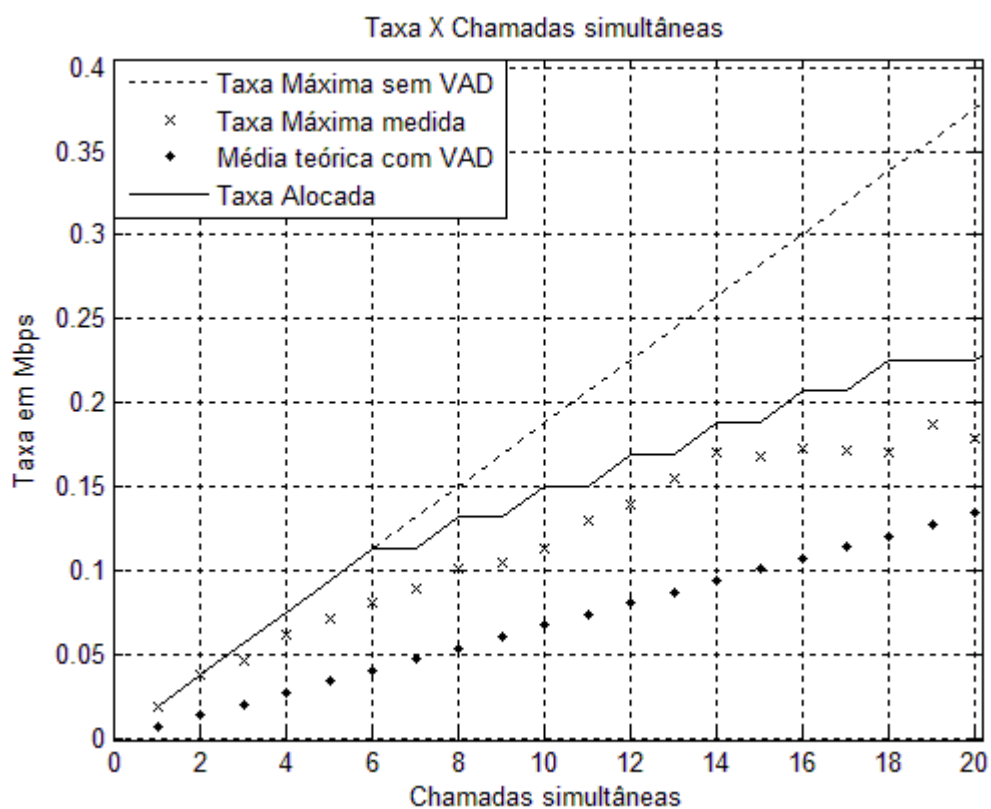


FIGURA 8 - Comparação entre as taxas medidas e as taxas calculadas para alocação para poucas chamadas

Comprovado o fato de que é possível prever a taxa máxima ocupada, o presente método será aplicado no capítulo a seguir em um enlace via satélite e seus efeitos serão avaliados a nível de percepção da qualidade de voz.

## 4 SIMULAÇÃO DE DESEMPENHO USANDO NS-2

### 4.1 Cenário 1 – Poucas chamadas por VSAT

A rede simulada pelo NS-2 com patch TDMA-DAMA [21] será composta por 20 RCSTs também conhecidos como VSATs (*Very Small Aperture Terminal*) conectados a Gateways de Acesso (*Access Gateways*) e a uma rede local de computadores (LAN). Todos os dados originados nos VSATs são encaminhados (via satélite geo-estacionário) ao Hub satelital que está conectado a um *backbone* terrestre com acesso à Internet e a um *Trunking Gateway* para interconexão à Rede de Telefonia Pública Comutada (RTPC) como mostrado na FIGURA 9.

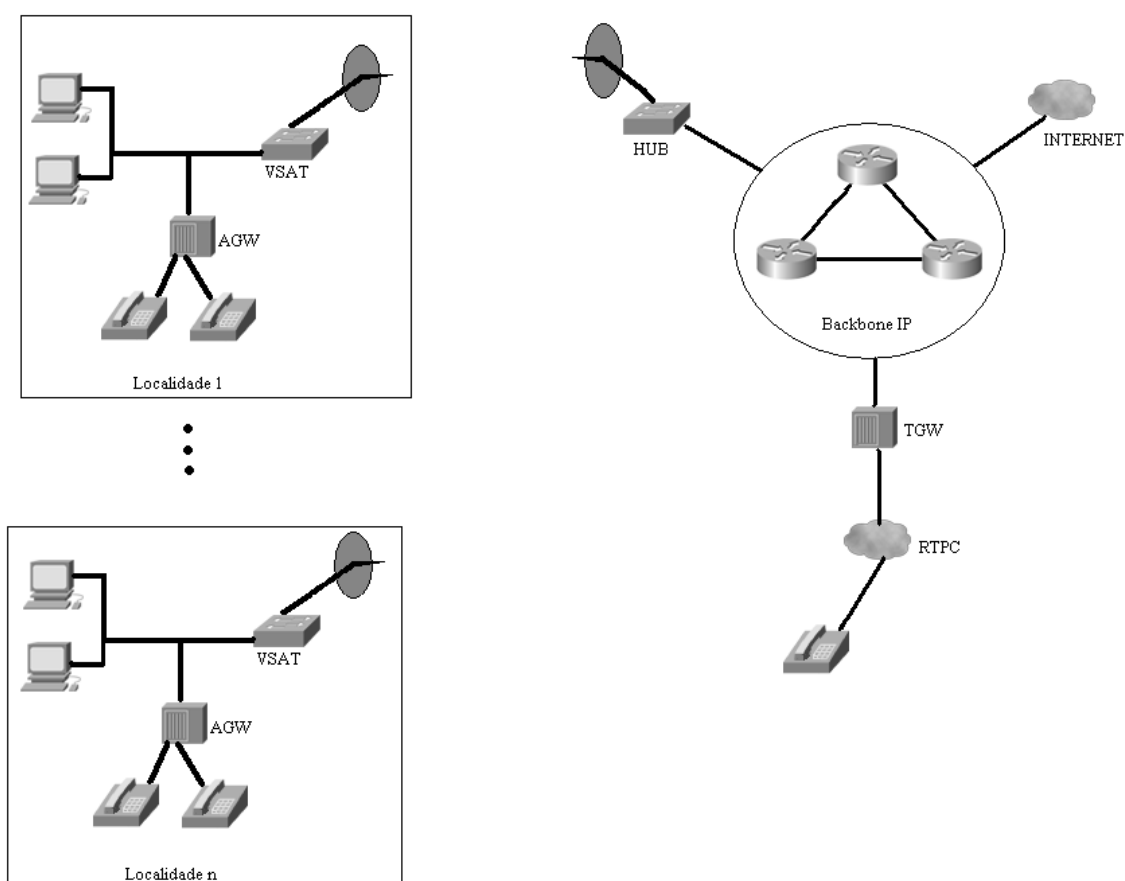


FIGURA 9 – Topologia para simulação

Em cada localidade há quatro elementos (além do VSAT) pertencentes à mesma LAN, ou seja, compartilham o meio de acesso. Um dos elementos estará fazendo uma transferência FTP com o provedor de Internet, simulando um *download*, durante todo o tempo da simulação. Outro elemento estará fazendo requisições HTTP a intervalos aleatórios numa distribuição exponencial com média de 5 segundos e estará recebendo dados de tamanho aleatório com distribuição exponencial e média de 1024 bytes [22], simulando uma navegação *Web*. Os outros dois elementos estarão numa comunicação bi-direcional de tráfego RTP, como se fossem duas chamadas VoIP em um mesmo *Access Gateway*. Essa configuração foi escolhida, pois representa um exemplo típico do uso de VoIP junto com tráfego *Web* e transferência de arquivos, muito comum na rede de Internet hoje.

A avaliação da qualidade de voz será feita pela medida do MOS (*Mean Opinion Score*) de acordo com o Modelo-E [3] através da análise de parâmetros de rede (retardo, *jitter*, perda de pacotes) observados no *Trunking Gateway* (as medições devem ser feitas nesse elemento pois é nessa direção que é utilizado o canal satelital de retorno).

O canal de retorno tem uma largura de banda de 2 Mbps compartilhada entre todos os terminais através de multiplexação temporal (TDMA) onde um quadro é composto de 95 janelas de tempo (94 úteis) de 106 bytes (duas células ATM por janela de tempo), resultando numa taxa de 21,05 kbps para cada janela de tempo e onde o superquadro é composto de três quadros. As requisições de banda podem ser feitas a cada superquadro, cuja duração é de 120,84 segundos. A largura de banda disponível a cada terminal é alocada dinamicamente através da atribuição de janelas de tempo conforme for requisitado.

O uso do patch TDMA-DAMA altera o comportamento normal do NS2 de forma a simular o ambiente DVB-RCS da seguinte maneira:

- Os dados que se encontram na fila de camada 2 do terminal só são transmitidos durante a sua janela de tempo;
- Se o pacote IP for maior que a janela de tempo, o pacote é fragmentado no terminal e depois reagrupado no hub satelital;

- Os dados de camada 2 provenientes da rede local são monitorados durante o período do superquadro, obtendo assim a taxa de bits de entrada nesse período. Se a taxa de bits de entrada for maior que a banda alocada o terminal solicita banda ao NCC, havendo banda disponível, ou seja, janelas de tempo disponíveis, o NCC disponibiliza a banda solicitada a partir do próximo superquadro. Se a taxa de bits é menor o terminal solicita desalocação e se for igual não solicita nada.

O fluxo RTP que será analisado equivale ao codec G.729 com período de packetização de 40 milissegundos e começa a trafegar no tempo 4 segundos, de forma que já se inicia com uma situação de tráfego no enlace local e termina no tempo 60 segundos. O receptor estará usando um *buffer* de *jitter* de 40 milissegundos para suavizar o efeito de *jitter* inerente a uma rede VoIP.

#### **4.1.1 Simulação com Esquema Padrão de Alocação sem VAD**

Para efeitos de comparação a primeira simulação corresponde ao cenário atualmente em uso, ou seja, VAD desativado e esquema padrão de alocação de banda (RBDC + CRA) sem transmissão de dados. O objetivo dessa simulação é demonstrar o nível de qualidade que se observa atualmente na rede. O uso do CRA garante que haja sempre pelo menos uma janela de tempo alocada. As atualizações das requisições RBDC são realizadas, quando necessário, a cada superquadro, ou seja, a cada 120,84 milissegundos. A FIGURA 10 apresenta o gráfico de tempo de atraso fim-a-fim de cada pacote, para uma chamada.

O primeiro pacote demora 272 milissegundos para chegar ao destino, e como todos os atrasos estão abaixo de 312 milissegundos não haverá descarte no *buffer* do receptor. A oscilação observada entre 250 e 285 milissegundos é devido ao retardo de acesso, pois há descasamento de tempo entre o período de packetização e o período da janela de tempo alocada.



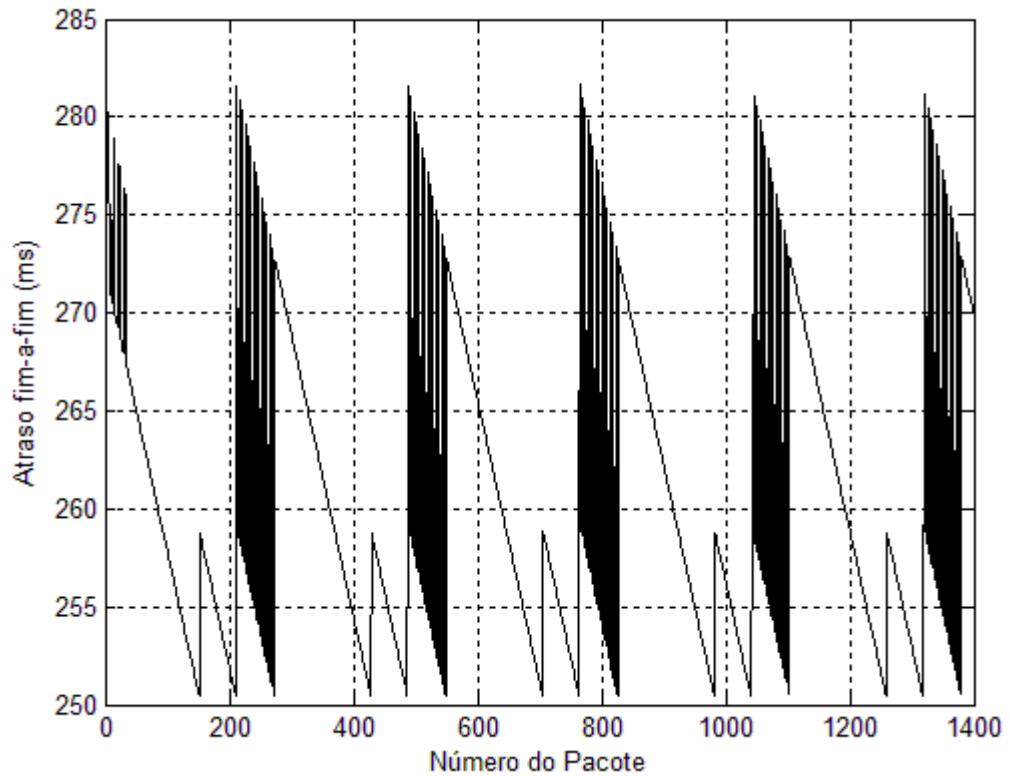


FIGURA 10 - Transmissão de voz com esquema padrão de alocação

O resultado é uma perda de 0% dos pacotes, o *jitter* medido foi de 2,8 milissegundos e a qualidade percebida na escala MOS pode ser obtida pelo Modelo-E da seguinte maneira, conforme descrito na seção 2.6:

1 – Cálculo do fator  $Id$

$$Ta = 40 + 272 + 40 = 352 ms$$

$$X = \frac{\lg\left(\frac{Ta}{100}\right)}{\lg 2} = 1,81$$

$$Id = 25 \left\{ \left(1 + X^6\right)^{\frac{1}{6}} - 3 \left(1 + \left[\frac{X}{3}\right]^6\right)^{\frac{1}{6}} + 2 \right\} = 20,023$$

## 2 – Cálculo do fator $I_{eef}$

Para o codec G.729 os valores de  $I_e$  e  $B_{pl}$  são 10 e 19 respectivamente

$$I_{eef} = I_e + (95 - I_e) \cdot \frac{P_{pl}}{P_{pl} + B_{pl}} = 10$$

## 3 – Fator A

Por se tratar de uma comunicação via satélite, pode-se determinar seu valor correspondente no QUADRO 2.

$$A = 20$$

## 4 – Cálculo do fator $R$

$$R = 93,2 - I_d - I_{eef} + A = 93,2 - 20,023 - 10 + 20 = 83,177$$

## 5 – Cálculo do $MOS$

$$MOS = 1 + 0,035R + R(R - 60)(100 - R)7 \cdot 10^{-6} = 4,14$$

Este valor corresponde, conforme QUADRO 1, a usuários satisfeitos, demonstrando que o cenário atualmente em uso apresenta um nível de qualidade de voz aceitável.

A seguir a mesma simulação, porém com tráfego de dados simultâneos para verificar como este afeta a qualidade percebida. O gráfico do tempo de atraso fim-a-fim pode ser visto na FIGURA 11.

O primeiro pacote demora 272 milissegundos para chegar ao destino, logo todos os pacotes com atraso maior que 312 milissegundos são descartados no receptor. Pode-se observar um pico de 350 milissegundos no início da transmissão, ocasionado pela falta de banda disponível nesse instante, logo após, mais uma janela de tempo é alocada para esse VSAT que pode ser percebido pela estabilização dos tempos de transmissão.

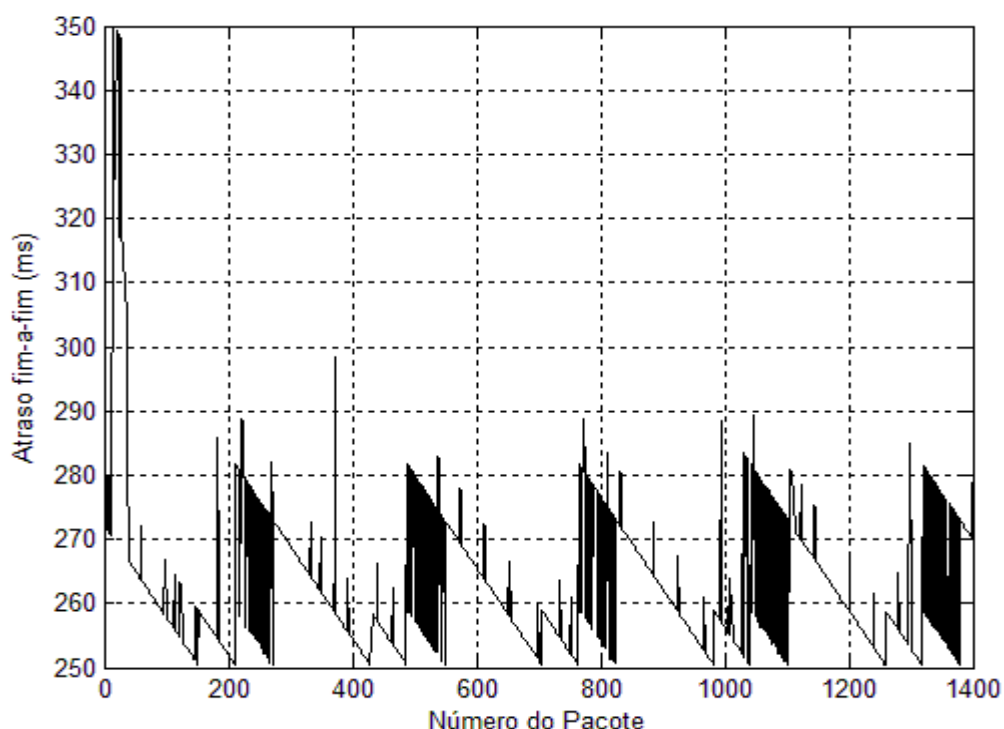


FIGURA 11 - Transmissão de voz e dados com esquema padrão de alocação

O resultado é uma perda de 1,42% dos pacotes, o *jitter* medido foi de 6,7 milissegundos e a qualidade percebida na escala MOS foi de 3,91 que corresponde a alguns usuários insatisfeitos, demonstrando que a transmissão simultânea de dados e voz degrada a qualidade de voz quando é usado o esquema padrão de alocação.

#### 4.1.2 Simulação com Esquema Padrão de Alocação com VAD

A próxima simulação demonstra os efeitos ao se ativar o VAD. O valor médio do tempo LIGADO é de 360 milissegundos e o valor médio do tempo DESLIGADO é de 650 milissegundos conforme os estudos de [9] e [14] e não há transmissão de dados.

Quando o tráfego está no estado LIGADO ocorre uma alocação da banda satelital e no estado DESLIGADO uma desalocação. Toda vez que uma alocação é necessária ocorre um aumento no retardo de acesso. Na FIGURA 12 é apresentado o gráfico do tempo de atraso fim-a-fim de cada pacote numa simulação onde há apenas tráfego de voz.

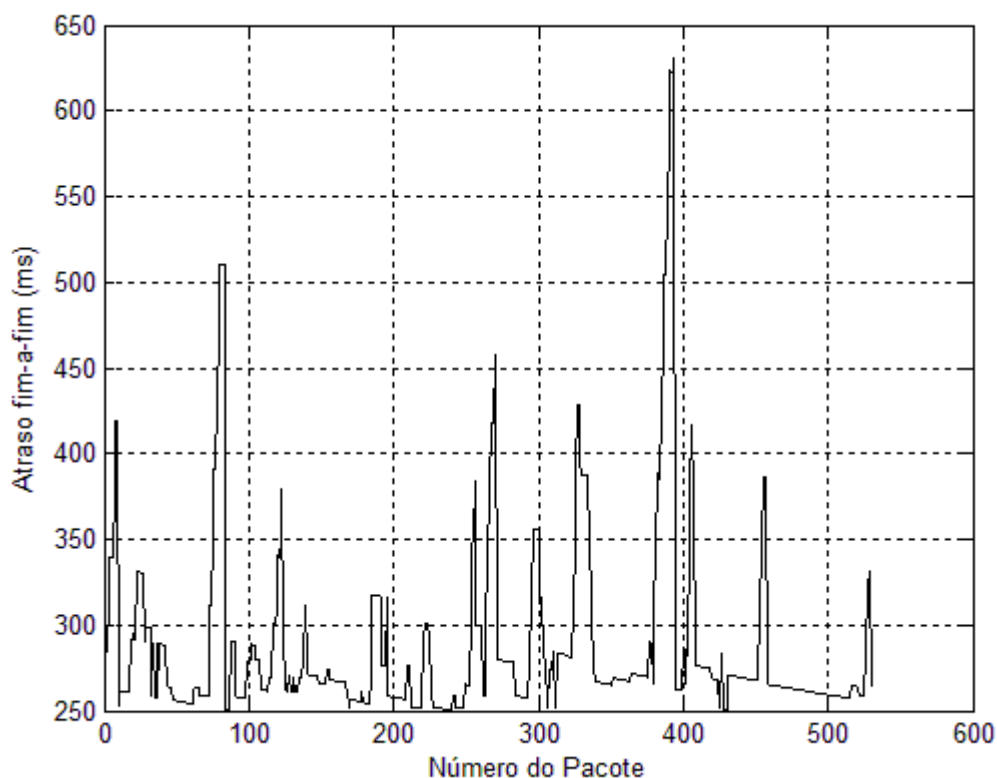


FIGURA 12 - Transmissão de voz com VAD usando esquema padrão de alocação

Como o primeiro pacote demorou 268 milissegundos para chegar ao receptor, todos os atrasos acima de 308 milissegundos são descartados pelo *buffer* do receptor causando uma maior perda de pacotes e conseqüentemente uma deterioração na qualidade percebida.

O resultado obtido foi de 17% de perdas, *jitter* de 8 milissegundos e a qualidade percebida foi de 2,22 na escala MOS, o que corresponde a todos os usuários insatisfeitos, portanto, mesmo havendo apenas tráfego de voz, a ativação do VAD causa uma grande degradação na qualidade de voz.

A FIGURA 13 mostra o gráfico para a simulação de voz com VAD simultaneamente à transmissão de dados usando o esquema padrão de alocação. O objetivo dessa simulação é verificar se o tráfego de dados alivia as transições de taxa ocasionado pelo VAD de forma que haja uma banda mais constante alocada durante a simulação.

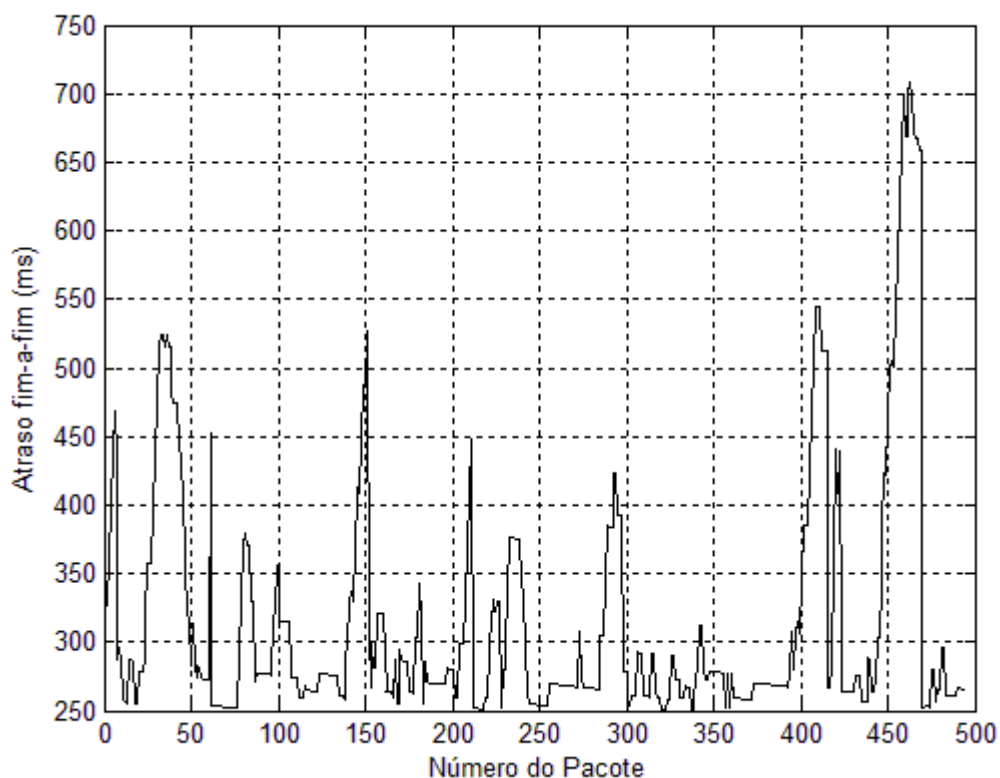


FIGURA 13 - Transmissão de voz com VAD e dados usando esquema padrão de alocação

Como o primeiro pacote demorou 281 milissegundos para chegar ao receptor, todos os atrasos acima de 321 milissegundos são descartados pelo *buffer* do receptor. O resultado obtido foi de 26% de perdas, *jitter* de 11,73 milissegundos e a qualidade percebida foi de 1,74 na escala MOS, o que corresponde a todos os usuários insatisfeitos, o que demonstra que o tráfego de dados simultâneo ao uso do VAD degrada mais ainda a qualidade de voz.

### 4.1.3 Simulação com Pré-Alocação de Banda com VAD

Alterando-se o algoritmo de requisição de banda usado pelo VSAT para o pior caso observado, ou seja, transmissão de voz com VAD e dados simultaneamente, de modo a pré-alocar a largura de banda no início de cada sessão RTP e ajustando-se a porcentagem de bloqueio para 3%, o resultado pode ser observado na FIGURA 14.

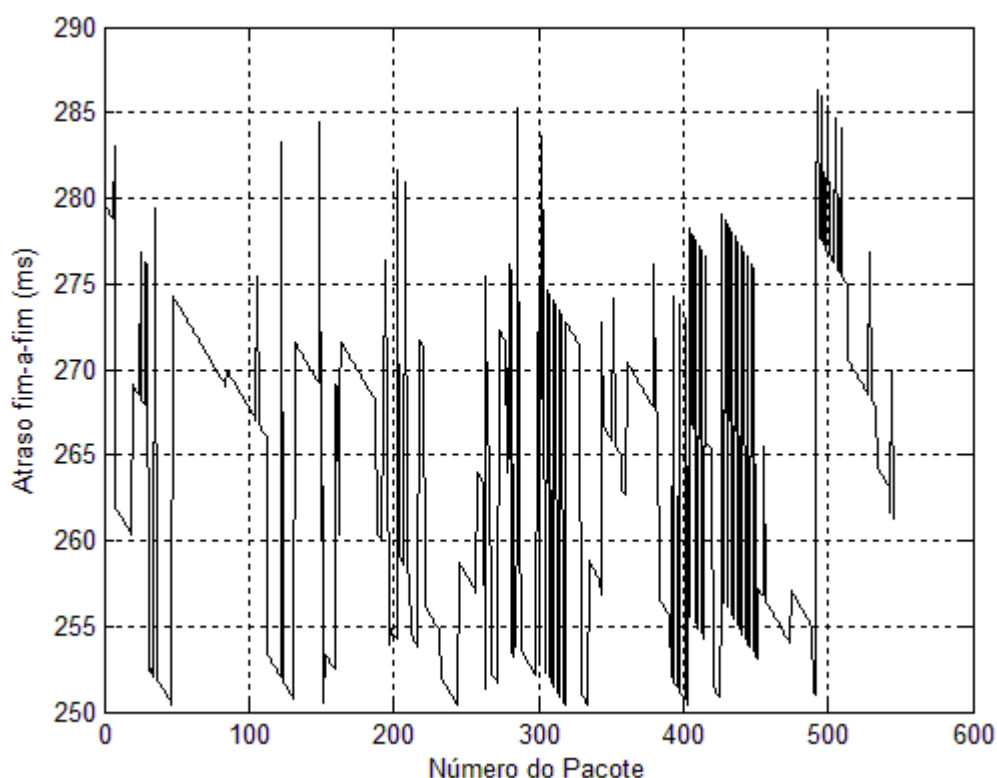


FIGURA 14 - Transmissão de voz com VAD e dados usando pré-alocação de banda

Observa-se que agora todos os atrasos estão abaixo de 290 milissegundos o que significa que praticamente não há formação de fila no VSAT, visto que há banda suficiente para encaminhar o tráfego entrante imediatamente. O índice de perdas foi de 0%, o *jitter* medido foi 2,06 milissegundos e o MOS foi 4,11, o que corresponde a usuários satisfeitos, resultado equivalente à primeira simulação, porém aqui há tráfego de dados e economia de banda.

#### 4.1.4 Resumo dos resultados do Cenário 1

Os resultados do cenário 1 no Quadro 3 demonstram que apenas duas situações apresentam uma qualidade satisfatória: A primeira é quando não se tem VAD e nem transmissão de dados simultâneos e usa o esquema padrão de alocação e a segunda situação é quando é usado o esquema de pré-alocação de banda, mesmo com VAD ativo e transmissão de dados simultâneos.

	<b>Perda</b>	<b>Jitter</b>	<b>MOS</b>
Voz sem VAD Alocação Padrão	0%	2,814 ms	4,1 – Usuários satisfeitos
Voz+Dados sem VAD Alocação Padrão	1,42%	6,697 ms	3,9 – Alguns usuários insatisfeitos
Voz com VAD Alocação Padrão	17,1%	8,075 ms	2,2 – Todos os usuários insatisfeitos
Voz+Dados com VAD Alocação Padrão	26%	11,743 ms	1,7 – Todos os usuários insatisfeitos
Voz+Dados com VAD Pré-Alocação	0%	2,062 ms	4,1 – Usuários satisfeitos

QUADRO 3 – Resultado das simulações para o cenário 1

#### 4.2 Cenário 2 – Várias chamadas por VSAT

O segundo cenário simulado apresenta as mesmas características VoIP (codec, período de paquetização, buffer *antijitter*), porém agora são 2 VSATs com 20 chamadas simultâneas em cada, concorrendo com tráfego de dados FTP e requisições HTTP. O objetivo deste cenário é avaliar o desempenho do método proposto em condições de tráfego mais intenso no VSAT.

#### 4.2.1 Simulação com Esquema Padrão de Alocação com VAD

Esta simulação demonstra qual o comportamento do tráfego VoIP com VAD em condição de alto tráfego e esquema de alocação padrão no VSAT.

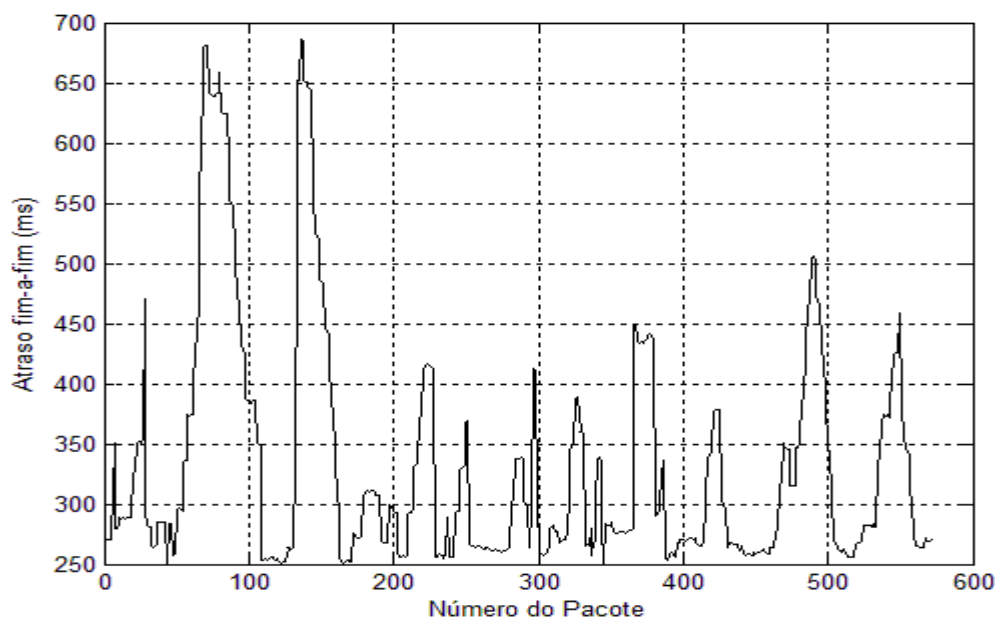


FIGURA 15 - Transmissão de voz com VAD e dados usando alocação padrão em alto tráfego

Como pode ser visto na FIGURA 15, a qualidade percebida é muito ruim. O primeiro pacote teve um atraso de 270 milissegundos e todos os pacotes com atraso superior a 310 milissegundos são descartados no receptor resultando em 30% de perdas. O *jitter* medido foi de 6,43 milissegundos e o MOS de 1,45 o que corresponde a todos os usuários insatisfeitos.

#### 4.2.2 Simulação com Pré-Alocação de Banda com VAD

Usando o algoritmo de pré-alocação para as mesmas condições de tráfego da simulação anterior o resultado pode ser observado na FIGURA 16.



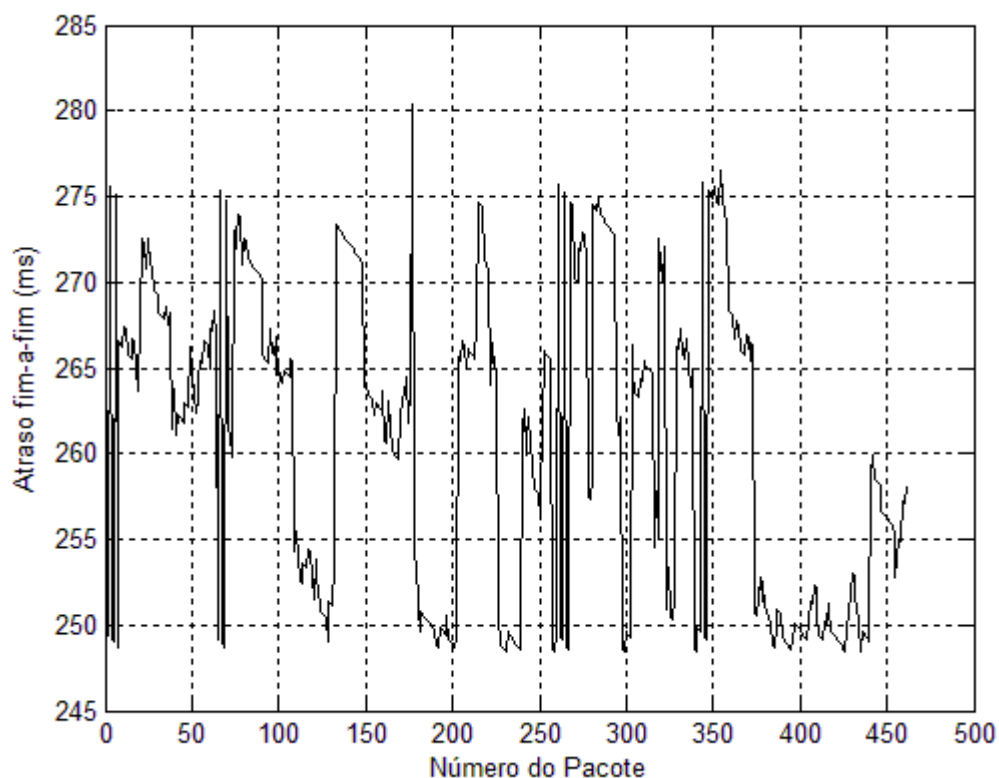


FIGURA 16 - Transmissão de voz com VAD e dados usando pré-alocação em alto tráfego

O atraso do primeiro pacote foi de 254 milissegundos e como não há pacotes com atraso superior a 294 milissegundos o índice de perdas é zero, o *jitter* observado foi de 0,9 milissegundos e o índice MOS é de 4,2 o que representa usuários satisfeitos com a qualidade percebida.

Os resultados demonstram que o método apresentado garante um nível aceitável de qualidade de voz tanto em condições de baixo tráfego quanto de alto tráfego.

### 4.2.3 Resumo dos resultados do Cenário 2

Os resultados do cenário 2 no Quadro 4 demonstram que mesmo em situação de alto tráfego com VAD ativo e transmissão simultânea de dados, o esquema de pré-alocação de banda atinge um nível satisfatório de qualidade de voz.

	<b>Perda</b>	<b>Jitter</b>	<b>MOS</b>
Voz+Dados com VAD Alocação Padrão	30%	6,43 ms	1,45 – Todos os usuários insatisfeitos
Voz+Dados com VAD Pré-Alocação	0%	0,9 ms	4,2 – Usuários satisfeitos

QUADRO 4 – Resultado das simulações para o cenário 2

## CONCLUSÃO

Para se levar telefonia a regiões remotas, o uso de VoIP sobre uma rede satelital usando DVB-RCS é uma opção, porém as redes atualmente em uso não se beneficiam do VAD devido à alta deterioração na qualidade de voz percebida. Fundamentando-se na teoria de filas, o esquema proposto resolve esse problema através da pré-alocação de banda, determinada estocasticamente e validada pelo simulador de redes NS-2, de modo a manter a qualidade de voz em um nível aceitável e ao mesmo tempo economizar largura de banda satelital.

Com base na Teoria de Filas, as equações obtidas através de uma modelagem a nível de pacote permitem prever a banda necessária com boa precisão, porém essas equações apresentam uma grande complexidade computacional, o que inviabiliza sua aplicação em tempo real. Porém uma modelagem a nível de chamadas resulta em equações mais simples que podem ser calculadas em tempo real, bastando para isso relacionar a probabilidade de bloqueio (nível de chamadas) com a qualidade de voz, que é fortemente dependente da probabilidade de perdas de pacotes decorrentes do tempo de espera excessivo na fila do terminal RCST (nível de pacote). O esquema resultante, portanto, deve ser aplicado em duas fases: A fase de Dimensionamento determina a probabilidade de bloqueio aceitável e a fase de Operação determina a banda a ser alocada dinamicamente em tempo real.

Dois cenários de rede satelital foram testados, baixo e alto tráfego de voz. O cenário de baixo tráfego é mais relevante, pois é esse o cenário encontrado nas regiões atendidas pelo PGMU e é onde trabalhos propostos anteriormente falham em prever banda e manter uma boa qualidade de voz ao mesmo tempo. Em ambos os cenários foram feitas comparações entre o esquema proposto e o esquema atualmente em uso nas redes DVB-RCS. Todas as simulações foram avaliadas pelo Modelo-E e os melhores resultados obtidos foram naquelas em que era feito pré-alocação de banda.

A dedução das fórmulas leva em conta o uso de *codecs* em que não há variação do tamanho do pacote durante a chamada, que é o caso dos *codecs* da família G.7xx padronizada pelo ITU-T e de uso muito comum nas operadoras de telefonia fixa. Também foi considerado o uso de um buffer *antijitter* de tamanho fixo, porém é comum o uso de buffer autoadaptativo que pode levar a um incremento no nível MOS ao, eventualmente, reduzir o fator  $Ta$ , portanto, os resultados apresentados apresentam uma visão pessimista sobre a qualidade de voz percebida, ou seja, em uma rede que faça uso dessa funcionalidade, a qualidade tende a ser melhor do que as apresentadas nas simulações.

Como trabalhos futuros propõe-se um estudo mais detalhado sobre alocação de banda via satélite para tráfegos TCP que possam ocorrer simultaneamente ou não ao tráfego de voz de modo a garantir um bom nível de QoS para ambos os tráfegos e uma implementação em um sistema real para se avaliar a eficiência do método descrito nesse trabalho.

## REFERÊNCIAS

- [1] F. H. Cardoso, “DECRETO N° 2.592, DE 15 DE MAIO DE 1998” – Presidência da República
- [2] L. I. L. da Silva, “DECRETO N° 4.769, DE 27 DE JUNHO DE 2003” – Presidência da República
- [3] ITU-T Rec. G.107 – The E-model, a computational model for use in transmission planning – May 2005
- [4] ITU-T Rec. P.800 – Methods for subjective determination of transmission quality – August 1996
- [5] ETSI EN 301 790 v1.3.1 – “Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems”, March – 2003
- [6] H. Skinnemoen, A. Vermesan, A. Iuoras, G. Adams, X. Lobao, “VoIP over DVB-RCS with QoS and Bandwidth on Demand”, IEEE Wireless Communications – October 2005
- [7] ITU-T Rec. G.1010 – End-user multimedia QoS categories – November 2001
- [8] RFC 1889 – RTP: A Transport Protocol for Real-Time Applications, January 1996
- [9] M. Kalama, G. Acar, B. G. Evans, S. Mourgues, “VoIP over DVB-RCS Satellite Systems: A novel Capacity Request Mechanism for improved Voice Activity Detection”, 978-1-4244-1645-5/08 IEEE 2008, pages 2957-2961.
- [10] V. A. Reguera, F. F. A. Paliza, W. Godoy Jr., E. M. G. Fernández, “On the impact of active queue management on VoIP quality of service”, Computer Communications 31 (2008) 73-87
- [11] A. Sathiaselan, G. Fairhurst, “Performance of VoIP using DCCP over a DVB-RCS satellite network”, In ICC’2007, IEEE International Conference on Communications, pages 13-18.
- [12] O. Alphand, P. Berthou, T. Gayraud, S. Combes, “QoS architecture over DVB-RCS satellite networks in a NGN framework”, In Globecom 2005 IEEE
- [13] M. Menth, A. Binzenhöfer, S. Mühleck, “Source Models for Speech Traffic Revisited”, IEEE/ACM Transactions on Networking, Vol 17, N° 4, August 2009

- [14] K. Sriram, "Characterizing Superposition Arrival Processes in Packet Multiplexers for Voice and Data", IEEE Journal on Selected Areas in Communications, Volume 4, Issue 6, Sep 1986 Page(s):833 – 846
- [15] R. B. Cooper, "Introduction to QUEUEING THEORY", Second Edition, Elsevier North Holland Inc. 1981
- [16] M. H. Ackroyd, "Computing the Waiting Time Distribution for the G/G/1 Queue by Signal Processing Methods", IEEE Transactions on Communications, Vol. Com-28, No. 1, January 1980
- [17] RFC 3261 – SIP: Session Initiation Protocol, June 2002
- [18] RFC 3015 – MEGACO Protocol Version 1.0, November 2000
- [19] RFC 2205 – Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification, September 1997
- [20] ITU-T Rec. G.729 - Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP), January 2007
- [21] R. Secchi, Tdma-dama: satellite on-demand bandwidth allocation for ns-2, <http://wnet.isti.cnr.it/software/tdma-dama>
- [22] K. Fall, K. Varadhan, "The ns Manual", The VINT Project, January 2009

## APÊNDICE A – Equação Integral de Lindley

Considerando-se um sistema de filas G/G/1 com disciplina de serviço FIFO deseja-se determinar a função densidade de probabilidade do tempo em que um pacote (ou cliente, na linguagem comumente usada na teoria de filas) aguarda na fila.

Sejam as seguintes variáveis aleatórias:

$Y_j$ , tempo entre as chegadas dos pacotes  $(j-1)$  e  $j$ ;

$X_j$ , tempo de serviço do pacote  $j$ ;

$G_j$ , tempo de espera (na fila) do pacote  $j$ ;

E definindo-se suas FDP como:

$$A(t) = P[Y_j \leq t] \quad (A.1)$$

$$B(t) = P[X_j \leq t] \quad (A.2)$$

$$W_j(t) = P[G_j \leq t] \quad (A.3)$$

As variáveis aleatórias  $\{Y_j\}$  e  $\{X_j\}$  são independentes e descritas por  $A(t)$  e  $B(t)$  independente do índice  $j$ .

As suas fdp são definidas como:

$$a(t) = \frac{dA(t)}{dt} \quad (A.4)$$

$$b(t) = \frac{dB(t)}{dt} \quad (A.5)$$

$$w_j(t) = \frac{dW_j(t)}{dt} \quad (A.6)$$

O tempo de espera do  $(j+1)$ -ésimo pacote pode ser dado por:

$$G_{j+1} = \begin{cases} G_j + X_j - Y_{j+1} & \text{se } G_j + X_j - Y_{j+1} \geq 0 \\ 0 & \text{se } G_j + X_j - Y_{j+1} < 0 \end{cases} \quad (A.7)$$

Pode-se aqui definir uma nova variável aleatória

$$U_j = X_j - Y_{j+1} \quad (\text{A.8})$$

$$C(t) = P[U_j = X_j - Y_{j+1} \leq t] \quad (\text{A.9})$$

$$c(t) = \int_0^{\infty} b(t + \tau) a(\tau) d\tau = a(-t) * b(t) \quad (\text{A.10})$$

Pode-se reescrever (A.7) como:

$$G_{j+1} = \begin{cases} G_j + U_j & \text{se } G_j + U_j \geq 0 \\ 0 & \text{se } G_j + U_j < 0 \end{cases} \quad (\text{A.11})$$

Para valores positivos de  $t$  sua fdp pode ser obtida pela convolução das fdp de  $G_j$  e  $U_j$

$$w_{j+1}(t) = w_j(t) * c(t) \quad t > 0 \quad (\text{A.12})$$

Devido ao caráter assimétrico da equação (A.11), todos os possíveis valores negativos da soma  $G_j + U_j$ , convergem para um tempo de espera igual a zero, portanto:

$$w_{j+1}(t) = \delta(t) \int_{-\infty}^{0^+} [w_j(t) * c(t)] dt \quad t = 0 \quad (\text{A.13})$$

Como o tempo de espera não pode ser negativo, sua fdp é nula para valores negativos de  $t$ :

$$w_{j+1}(t) = 0 \quad t < 0 \quad (\text{A.14})$$

Portanto, reunindo os intervalos, tem-se a equação integral de Lindley:

$$w_{j+1}(t) = \begin{cases} w_j(t) * c(t), & t > 0 \\ \delta(t) \int_{-\infty}^{0^+} [w_j(t) * c(t)] dt, & t = 0 \\ 0, & t < 0 \end{cases} \quad (\text{A.15})$$



## APÊNDICE B – Valores Padrão do Modelo-E

O quadro a seguir é o mesmo encontrado na recomendação G.107 e expressa os valores padrão e os limites para os vários parâmetros do Modelo-E.

Parameter	Abbr.	Unit	Default value	Permitted range	Remark
Send Loudness Rating	SLR	dB	+8	0 ... +18	Note 1
Receive Loudness Rating	RLR	dB	+2	-5 ... +14	Note 1
Sidetone Masking Rating	STMR	dB	15	10 ... 20	Note 2
Listener Sidetone Rating	LSTR	dB	18	13 ... 23	Note 2
D-Value of Telephone, Send Side	Ds	–	3	-3 ... +3	Note 2
D-Value of Telephone Receive Side	Dr	–	3	-3 ... +3	Note 2
Talker Echo Loudness Rating	TELR	dB	65	5 ... 65	
Weighted Echo Path Loss	WEPL	dB	110	5 ... 110	
Mean one-way Delay of the Echo Path	T	msec	0	0 ... 500	
Round Trip Delay in a 4-wire Loop	Tr	msec	0	0 ... 1000	
Absolute Delay in echo-free Connections	Ta	msec	0	0 ... 500	
Number of Quantization Distortion Units	qdu	–	1	1 ... 14	
Equipment Impairment Factor	Ie	–	0	0 ... 40	
Circuit Noise referred to 0 dBr-point	Nc	dBm0p	-70	-80 ... -40	
Noise Floor at the Receive Side	Nfor	dBmp	-64	–	Note 3
Room Noise at the Send Side	Ps	dB(A)	35	35 ... 85	
Room Noise at the Receive Side	Pr	dB(A)	35	35 ... 85	
Advantage Factor	A	–	0	0 ... 20	
NOTE 1 – Total values between microphone or receiver and 0 dBr-point.					
NOTE 2 – Fixed relation: $LSTR = STMR + D$ .					
NOTE 3 – Currently under study.					

## APÊNDICE C – SCRIPTS NS-2

### C.1 Voz sem VAD com Esquema Padrão

```
#####
#
#       Variable Settings
#
#####

# Skyplex parameters
Mac/Tdma set max_slot_num_ 94           ;# slot per frame (preamble excluded)
Mac/Tdma set slot_packet_len_ 106        ;# slot size (bytes)
Mac/Tdma set num_frame_ 3                ;# multiframe size (in frames)
Mac/Tdma set bandwidth_ 2Mb
Allocator/Proportional set delay_ 1.4

set EPS 0.000001
set MAX_ACTIVE_NODE 64

LL/Sat set bandwidth_ 2Mb

global ns
set ns [new Simulator]

set name onlyrtmp

set type RBDC

# Note: Even though "Static" is normally reserved for static
#       topologies, the satellite code will trigger a recalculation
#       of the routing tables whenever the topology changes.
#       Therefore, it is not so much "static" as "omniscient", in that
#       topology changes are known instantly throughout the topology.
#       See documentation for discussion of dynamic routing protocols.
$ns rtproto Static

#####
# Global configuration parameters
#####

global opt
set opt(chan) channel/Sat
set opt(bw_down) 8Mb; # Downlink bandwidth (satellite to ground)
set opt(bw_up) 2Mb; # Uplink bandwidth
set opt(phy) Phy/Sat
set opt(mac) Mac/Tdma
set opt(ifq) Queue/DropTail
set opt(qlim) 1000
set opt(ll) LL/Sat
set opt(wiredRouting) ON

# IMPORTANT This tracing enabling (trace-all) must precede link and node
# creation. Then following all node, link, and error model
# creation, invoke "$ns trace-all-satlinks $outfile"
set outfile [open $name.tr w]
$ns trace-all $outfile

set ev_file [open $name.out w]

#####
# Set up satellite and terrestrial nodes
#####

# GEO satellite: above Brazil -- lets put it at 61 deg. W (Satelite Amazonas)
$ns node-config -satNodeType geo-repeater \
               -phyType Phy/Repeater \
               -channelType $opt(chan) \
               -downlinkBW $opt(bw_down) \
               -wiredRouting $opt(wiredRouting)

set nodeSat [$ns node]
$nodeSat set-position -61
```

```

# Terminals:
$ns node-config -satNodeType terminal
set nodeHub [$ns node]
set nodeTerm(1) [$ns node]
set nodeTerm(2) [$ns node]
set nodeTerm(3) [$ns node]
set nodeTerm(4) [$ns node]
set nodeTerm(5) [$ns node]
set nodeTerm(6) [$ns node]
set nodeTerm(7) [$ns node]
set nodeTerm(8) [$ns node]
set nodeTerm(9) [$ns node]
set nodeTerm(10) [$ns node]
set nodeTerm(11) [$ns node]
set nodeTerm(12) [$ns node]
set nodeTerm(13) [$ns node]
set nodeTerm(14) [$ns node]
set nodeTerm(15) [$ns node]
set nodeTerm(16) [$ns node]
set nodeTerm(17) [$ns node]
set nodeTerm(18) [$ns node]
set nodeTerm(19) [$ns node]
set nodeTerm(20) [$ns node]

$nodeHub set-position -22.9 -43.2;      # Rio de Janeiro
$nodeTerm(1) set-position -1.5 -48.5 ;# Belém
$nodeTerm(2) set-position -19.9 -43.9 ;# Belo Horizonte
$nodeTerm(3) set-position 2.8 -60.7 ;# Boa Vista
$nodeTerm(4) set-position -15.8 -47.9 ;# Brasília
$nodeTerm(5) set-position -20.4 -54.6 ;# Campo Grande
$nodeTerm(6) set-position -15.6 -56.1 ;# Cuiabá
$nodeTerm(7) set-position -25.4 -49.3 ;# Curitiba
$nodeTerm(8) set-position -27.6 -48.5 ;# Florianópolis
$nodeTerm(9) set-position -3.7 -38.5 ;# Fortaleza
$nodeTerm(10) set-position 0.0 -51.1 ;# Macapá
$nodeTerm(11) set-position -9.7 -35.7 ;# Maceió
$nodeTerm(12) set-position -3.1 -60.0 ;# Manaus
$nodeTerm(13) set-position -30.0 -51.2 ;# Porto Alegre
$nodeTerm(14) set-position -8.8 -63.9 ;# Porto Velho
$nodeTerm(15) set-position -10.0 -67.8 ;# Rio Branco
$nodeTerm(16) set-position -13.0 -38.5 ;# Salvador
$nodeTerm(17) set-position -2.5 -44.3 ;# São Luís
$nodeTerm(18) set-position -23.5 -46.6 ;# São Paulo
$nodeTerm(19) set-position -5.1 -42.8 ;# Teresina
$nodeTerm(20) set-position -20.3 -40.3 ;# Vitória

#####
# Set up links
#####

# GSIs to the geo satellite:
$nodeHub add-gsl geo $opt(l1) $opt(ifq) $opt(qlim) $opt(mac) $opt(bw_up) \
    $opt(phy) [$nodeSat set downlink_] [$nodeSat set uplink_]
set req(0) [$nodeHub install-requester Requester/RBDC]
$nodeHub install-allocator Allocator/Proportional
#$nodeHub insert-sat-ttl
$nodeHub trace-event $ev_file

for {set i 1} {$i <= 20} {incr i} {
    $nodeTerm($i) add-gsl geo $opt(l1) $opt(ifq) $opt(qlim) $opt(mac) $opt(bw_up) \
        $opt(phy) [$nodeSat set downlink_] [$nodeSat set uplink_]
    set req($i) [$nodeTerm($i) install-requester Requester/RBDC]
    # $nodeTerm($i) insert-sat-ttl
}

#####
# Set up Core
#####

$ns node-config -reset
set nodeCore [$ns node]
set nodeMgw1 [$ns node]
set nodeMgw2 [$ns node]
set nodewww [$ns node]

$ns duplex-link $nodeHub $nodeCore 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodeMgw1 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodeMgw2 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodewww 50Mb 2ms DropTail

#####
# Set up Remote Nodes
#####

```

```

set initialnode 27
set numberVSATs 20 ;
for {set lannumber 1} {$lannumber <= $numberVSATs} {incr lannumber} {
    set lan($lannumber) [$ns newLan "" 0.5Mb 2ms LL Queue/DropTail MAC/802_3 Channel]
    for {set nodenumber [expr $initialnode + 5*($lannumber -1)]} {$nodenumber < [expr
$initialnode + 5*($lannumber -1)] +4} {incr nodenumber} {
        set n($nodenumber) [$ns node]
        $lan($lannumber) addNode "$n($nodenumber)" 0.5Mb 2ms
    }
    $lan($lannumber) addNode "$nodeTerm($lannumber)" 0.5Mb 2ms
}

#####
# Tracing
#####
$ns trace-all-satlinks $outfile
$nodeTerm(1) trace-event $ev_file

#####
# Set up RTP params
#####

set packetSize 94; # Value in bytes
set ptime 0.04 ; # Value in seconds

#####
# Attach agents
# First Lan
#####

    set lannumber 1

    set nodenumber [expr $initialnode + 5*($lannumber -1)]

    set s($nodenumber) [new Agent/RTP]
    $ns attach-agent $n($nodenumber) $s($nodenumber)
    $s($nodenumber) set packetSize_ $packetSize
    $s($nodenumber) set interval_ $ptime
    set f($nodenumber) [new Agent/RTP]
    $ns attach-agent $nodeMgw1 $f($nodenumber)
    $f($nodenumber) set packetSize_ $packetSize
    $f($nodenumber) set interval_ $ptime
    $ns connect $s($nodenumber) $f($nodenumber)
    $ns at 4 "$s($nodenumber) start"
    $ns at 4 "$f($nodenumber) start"
    $ns at 60 "$s($nodenumber) stop"
    $ns at 60 "$f($nodenumber) stop"

    set secondnode [expr $nodenumber + 1]
    set s($secondnode) [new Agent/RTP]
    $ns attach-agent $n($secondnode) $s($secondnode)
    $s($secondnode) set packetSize_ $packetSize
    $s($secondnode) set interval_ $ptime
    set f($secondnode) [new Agent/RTP]
    $ns attach-agent $nodeMgw2 $f($secondnode)
    $f($secondnode) set packetSize_ $packetSize
    $f($secondnode) set interval_ $ptime
    $ns connect $s($secondnode) $f($secondnode)
    $ns at 0 "$s($secondnode) start"
    $ns at 0 "$f($secondnode) start"
    $ns at 60 "$s($secondnode) stop"
    $ns at 60 "$f($secondnode) stop"

#####
# Attach agents
# Subsequent Lans
#####

for {set lannumber 2} {$lannumber <= $numberVSATs} {incr lannumber} {

    set nodenumber [expr $initialnode + 5*($lannumber -1)]

    set s($nodenumber) [new Agent/RTP]
    $ns attach-agent $n($nodenumber) $s($nodenumber)
    $s($nodenumber) set packetSize_ $packetSize
    $s($nodenumber) set interval_ $ptime
    set f($nodenumber) [new Agent/RTP]
    $ns attach-agent $nodeMgw2 $f($nodenumber)
    $f($nodenumber) set packetSize_ $packetSize
    $f($nodenumber) set interval_ $ptime
    $ns connect $s($nodenumber) $f($nodenumber)
    $ns at 0 "$s($nodenumber) start"

```

```

$ns at 0 "$f($nodenumber) start"
$ns at 60 "$s($nodenumber) stop"
$ns at 60 "$f($nodenumber) stop"

set secondnode [expr $nodenumber + 1]
set s($secondnode) [new Agent/RTP]
$ns attach-agent $n($secondnode) $s($secondnode)
$s($secondnode) set packetSize_ $packetSize
$s($secondnode) set interval_ $ptime
set f($secondnode) [new Agent/RTP]
$ns attach-agent $nodeMgw2 $f($secondnode)
$f($secondnode) set packetSize_ $packetSize
$f($secondnode) set interval_ $ptime
$ns connect $s($secondnode) $f($secondnode)
$ns at 0 "$s($secondnode) start"
$ns at 0 "$f($secondnode) start"
$ns at 60 "$s($secondnode) stop"
$ns at 60 "$f($secondnode) stop"
}

#####
# Satellite routing                                     #
#####

set satrouteobject_ [new SatRouteObject]
$satrouteobject_ compute_routes

$ns at 70 "finish"

proc finish {} {
    global ns outfile
    $ns flush-trace
    close $outfile
    $ns halt
}

$ns run

```

## C.2 Voz sem VAD + Dados com Esquema Padrão

```

#####
#                                     #
#           Variable Settings         #
#                                     #
#####

# Skyplex parameters
Mac/Tdma set max_slot_num_ 94          ;# slot per frame (preamble excluded)
Mac/Tdma set slot_packet_len_ 106      ;# slot size (bytes)
Mac/Tdma set num_frame_ 3              ;# multiframe size (in frames)
Mac/Tdma set bandwidth_ 2Mb
Allocator/Proportional set delay_ 1.4

set EPS 0.000001
set MAX_ACTIVE_NODE 64

LL/Sat set bandwidth_ 2Mb

global ns
set ns [new Simulator]

set name satrt5

set type RBDC

# Note: Even though "Static" is normally reserved for static
#       topologies, the satellite code will trigger a recalculation
#       of the routing tables whenever the topology changes.
#       Therefore, it is not so much "static" as "omniscient", in that
#       topology changes are known instantly throughout the topology.
#       See documentation for discussion of dynamic routing protocols.
$ns rtproto Static

#####
# Global configuration parameters      #
#####

```

```

global opt
set opt(chan)          Channel/Sat
set opt(bw_down)       8Mb; # Downlink bandwidth (satellite to ground)
set opt(bw_up)         2Mb; # Uplink bandwidth
set opt(phy)           Phy/Sat
set opt(mac)           Mac/Tdma
set opt(ifq)           Queue/DropTail
set opt(qlim)          1000
set opt(ll)            LL/Sat
set opt(wiredRouting) ON

# IMPORTANT This tracing enabling (trace-all) must precede link and node
#             creation. Then following all node, link, and error model
#             creation, invoke "$ns trace-all-satlinks $outfile"
set outfile [open $name.tr w]
$ns trace-all $outfile

set ev_file [open $name.out w]

#####
# Set up satellite and terrestrial nodes
#####

# GEO satellite:  above Brazil -- lets put it at 61 deg. W (Satelite Amazonas)
$ns node-config -satNodeType geo-repeater \
    -phyType Phy/Repeater \
    -channelType $opt(chan) \
    -downlinkBw $opt(bw_down) \
    -wiredRouting $opt(wiredRouting)

set nodesSat [$ns node]
$nodesSat set-position -61

# Terminals:
$ns node-config -satNodeType terminal
set nodeHub [$ns node]
set nodeTerm(1) [$ns node]
set nodeTerm(2) [$ns node]
set nodeTerm(3) [$ns node]
set nodeTerm(4) [$ns node]
set nodeTerm(5) [$ns node]
set nodeTerm(6) [$ns node]
set nodeTerm(7) [$ns node]
set nodeTerm(8) [$ns node]
set nodeTerm(9) [$ns node]
set nodeTerm(10) [$ns node]
set nodeTerm(11) [$ns node]
set nodeTerm(12) [$ns node]
set nodeTerm(13) [$ns node]
set nodeTerm(14) [$ns node]
set nodeTerm(15) [$ns node]
set nodeTerm(16) [$ns node]
set nodeTerm(17) [$ns node]
set nodeTerm(18) [$ns node]
set nodeTerm(19) [$ns node]
set nodeTerm(20) [$ns node]

$nodeHub set-position -22.9 -43.2; # Rio de Janeiro
$nodeTerm(1) set-position -1.5 -48.5 ;# Belém
$nodeTerm(2) set-position -19.9 -43.9 ;# Belo Horizonte
$nodeTerm(3) set-position 2.8 -60.7 ;# Boa Vista
$nodeTerm(4) set-position -15.8 -47.9 ;# Brasília
$nodeTerm(5) set-position -20.4 -54.6 ;# Campo Grande
$nodeTerm(6) set-position -15.6 -56.1 ;# Cuiabá
$nodeTerm(7) set-position -25.4 -49.3 ;# Curitiba
$nodeTerm(8) set-position -27.6 -48.5 ;# Florianópolis
$nodeTerm(9) set-position -3.7 -38.5 ;# Fortaleza
$nodeTerm(10) set-position 0.0 -51.1 ;# Macapá
$nodeTerm(11) set-position -9.7 -35.7 ;# Maceió
$nodeTerm(12) set-position -3.1 -60.0 ;# Manaus
$nodeTerm(13) set-position -30.0 -51.2 ;# Porto Alegre
$nodeTerm(14) set-position -8.8 -63.9 ;# Porto Velho
$nodeTerm(15) set-position -10.0 -67.8 ;# Rio Branco
$nodeTerm(16) set-position -13.0 -38.5 ;# Salvador
$nodeTerm(17) set-position -2.5 -44.3 ;# São Luís
$nodeTerm(18) set-position -23.5 -46.6 ;# São Paulo
$nodeTerm(19) set-position -5.1 -42.8 ;# Teresina
$nodeTerm(20) set-position -20.3 -40.3 ;# Vitória

#####
# Set up links
#####

```

```

# GSLS to the geo satellite:
$nodeHub add-gsl geo $opt(l1) $opt(ifq) $opt(qlim) $opt(mac) $opt(bw_up) \
    $opt(phy) [$nodeSat set downlink_] [$nodeSat set uplink_]
set req(0) [$nodeHub install-requester Requester/RBDC]
$nodeHub install-allocator Allocator/Proportional
#$nodeHub insert-sat-ttl
$nodeHub trace-event $ev_file

for {set i 1} {$i <= 20} {incr i} {
    $nodeTerm($i) add-gsl geo $opt(l1) $opt(ifq) $opt(qlim) $opt(mac) $opt(bw_up) \
        $opt(phy) [$nodeSat set downlink_] [$nodeSat set uplink_]
    set req($i) [$nodeTerm($i) install-requester Requester/RBDC]
    # $nodeTerm($i) insert-sat-ttl
}

#####
# Set up Core
#####

$ns node-config -reset
set nodeCore [$ns node]
set nodeMgw1 [$ns node]
set nodeMgw2 [$ns node]
set nodewww [$ns node]

$ns duplex-link $nodeHub $nodeCore 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodeMgw1 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodeMgw2 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodewww 50Mb 2ms DropTail

#####
# Set up Remote Nodes
#####
set initialnode 27
set numberVSATS 20 ;
for {set lannumber 1} {$lannumber <= $numberVSATS} {incr lannumber} {
    set lan($lannumber) [$ns newLan "" 0.5Mb 2ms LL Queue/DropTail MAC/802_3 Channel]
    for {set nodenumber [expr $initialnode + 5*($lannumber -1)]} {$nodenumber < [expr
$initialnode + 5*($lannumber -1)] +4} {incr nodenumber} {
        set n($nodenumber) [$ns node]
        $lan($lannumber) addNode "$n($nodenumber)" 0.5Mb 2ms
    }
    $lan($lannumber) addNode "$nodeTerm($lannumber)" 0.5Mb 2ms
}

#####
# Tracing
#####
$ns trace-all-satlinks $outfile
$nodeTerm(1) trace-event $ev_file

#####
# Set up HTTP
#####
# Create page pool as a central page generator. Use PagePool/Math
set pgp [new PagePool/Math]
set tmp [new RandomVariable/Constant] ;## Page size generator
$tmp set val_ 1024 ;## average page size
$pgp ranvar-size $tmp

set tmp [new RandomVariable/Exponential] ;## Page age generator
$tmp set avg_ 5 ;## average page age
$pgp ranvar-age $tmp

set server [new Http/Server $ns $nodewww] ;## Create a server and link it to the central page
pool
$server set-page-generator $pgp

set cache [new Http/Cache $ns $nodeCore] ;## Create a cache
$cache connect $server

#####
# Set up RTP params
#####
set packetSize 94; # value in bytes
set ptime 0.04 ; # value in seconds

#####
# Attach agents
#
# First Lan
#####

```

```

set lannumber 1

set nodenumber [expr $initialnode + 5*($lannumber -1)]

set s($nodenumber) [new Agent/RTP]
$ns attach-agent $n($nodenumber) $s($nodenumber)
$s($nodenumber) set packetSize_ $packetSize
$s($nodenumber) set interval_ $ptime
set f($nodenumber) [new Agent/RTP]
$ns attach-agent $nodeMgw1 $f($nodenumber)
$f($nodenumber) set packetSize_ $packetSize
$f($nodenumber) set interval_ $ptime
$ns connect $s($nodenumber) $f($nodenumber)
$ns at 4 "$s($nodenumber) start"
$ns at 4 "$f($nodenumber) start"
$ns at 60 "$s($nodenumber) stop"
$ns at 60 "$f($nodenumber) stop"

set secondnode [expr $nodenumber + 1]
set s($secondnode) [new Agent/RTP]
$ns attach-agent $n($secondnode) $s($secondnode)
$s($secondnode) set packetSize_ $packetSize
$s($secondnode) set interval_ $ptime
set f($secondnode) [new Agent/RTP]
$ns attach-agent $nodeMgw2 $f($secondnode)
$f($secondnode) set packetSize_ $packetSize
$f($secondnode) set interval_ $ptime
$ns connect $s($secondnode) $f($secondnode)
$ns at 0 "$s($secondnode) start"
$ns at 0 "$f($secondnode) start"
#$ns at [expr 0.3 + ($lannumber - 1)] "$s($secondnode) start"
#$ns at [expr 0.3 + ($lannumber - 1)] "$f($secondnode) start"
$ns at 60 "$s($secondnode) stop"
$ns at 60 "$f($secondnode) stop"

set thirdnode [expr $nodenumber + 2]
set client($lannumber) [new Http/Client $ns $n($thirdnode)] ;## Create a client
set tmp [new RandomVariable/Exponential] ;## Poisson process as request sequence
$tmp set avg_ 5 ;## average request interval
$client($lannumber) set-interval-generator $tmp
$client($lannumber) set-page-generator $pgp
$client($lannumber) connect $cache
$ns at 0.0 "$client($lannumber) start-session $cache $server"

set fourthnode [expr $nodenumber + 3]
set tcp($lannumber) [new Agent/TCP]
set tcpsink($lannumber) [new Agent/TCPSink]
$ns attach-agent $nodewww $tcp($lannumber)
$ns attach-agent $n($fourthnode) $tcpsink($lannumber)
$ns connect $tcp($lannumber) $tcpsink($lannumber)
set ftp($lannumber) [new Application/FTP]
$ftp($lannumber) attach-agent $tcp($lannumber)
$ns at 0 "$ftp($lannumber) start"
#$ns at [expr 0.5 + ($lannumber - 1)] "$ftp($lannumber) start"
$ns at 60 "$ftp($lannumber) stop"

#####
# Attach agents #
# Subsequent Lans #
#####

for {set lannumber 2} {$lannumber <= $numberVSATs} {incr lannumber} {

    set nodenumber [expr $initialnode + 5*($lannumber -1)]

    set s($nodenumber) [new Agent/RTP]
    $ns attach-agent $n($nodenumber) $s($nodenumber)
    $s($nodenumber) set packetSize_ $packetSize
    $s($nodenumber) set interval_ $ptime
    set f($nodenumber) [new Agent/RTP]
    $ns attach-agent $nodeMgw2 $f($nodenumber)
    $f($nodenumber) set packetSize_ $packetSize
    $f($nodenumber) set interval_ $ptime
    $ns connect $s($nodenumber) $f($nodenumber)
    $ns at 0 "$s($nodenumber) start"
    $ns at 0 "$f($nodenumber) start"
    $ns at 60 "$s($nodenumber) stop"
    $ns at 60 "$f($nodenumber) stop"

    set secondnode [expr $nodenumber + 1]
    set s($secondnode) [new Agent/RTP]
    $ns attach-agent $n($secondnode) $s($secondnode)

```



```

    $s($secondnode) set packetSize_ $packetSize
    $s($secondnode) set interval_ $ptime
    set f($secondnode) [new Agent/RTP]
    $ns attach-agent $nodeMgw2 $f($secondnode)
    $f($secondnode) set packetSize_ $packetSize
    $f($secondnode) set interval_ $ptime
    $ns connect $s($secondnode) $f($secondnode)
    $ns at 0 "$s($secondnode) start"
    $ns at 0 "$f($secondnode) start"
    $ns at 60 "$s($secondnode) stop"
    $ns at 60 "$f($secondnode) stop"

    set thirdnode [expr $nodenumber + 2]
    set client($lannumber) [new Http/Client $ns $n($thirdnode)] ;## Create a client
    set tmp [new RandomVariable/Exponential] ;## Poisson process as request sequence
    $tmp set avg_ 5 ;## average request interval
    $client($lannumber) set-interval-generator $tmp
    $client($lannumber) set-page-generator $pgp
    $client($lannumber) connect $cache
    $ns at 0.0 "$client($lannumber) start-session $cache $server"

    set fourthnode [expr $nodenumber + 3]
    set tcp($lannumber) [new Agent/TCP]
    set tcpsink($lannumber) [new Agent/TCPSink]
    $ns attach-agent $nodewww $tcp($lannumber)
    $ns attach-agent $n($fourthnode) $tcpsink($lannumber)
    $ns connect $tcp($lannumber) $tcpsink($lannumber)
    set ftp($lannumber) [new Application/FTP]
    $ftp($lannumber) attach-agent $tcp($lannumber)
    $ns at 0 "$ftp($lannumber) start"
    $ns at 60 "$ftp($lannumber) stop"
}

#####
# Satellite routing                                     #
#####

set satrouteobject_ [new SatRouteObject]
$satrouteobject_ compute_routes

$ns at 70 "finish"

proc finish {} {
    global ns outfile
    $ns flush-trace
    close $outfile
    $ns halt
}

$ns run

```

### C.3 Voz com VAD e Esquema Padrão

```

#####
#                                     #
#           Variable Settings         #
#                                     #
#####

# Skyplex parameters
Mac/Tdma set max_slot_num_ 94          ;# slot per frame (preamble excluded)
Mac/Tdma set slot_packet_len_ 106      ;# slot size (bytes)
Mac/Tdma set num_frame_ 3              ;# multiframe size (in frames)
Mac/Tdma set bandwidth_ 2Mb
Allocator/Proportional set delay_ 1.4

set EPS 0.000001
set MAX_ACTIVE_NODE 64

LL/Sat set bandwidth_ 2Mb

global ns
set ns [new simulator]

set name onlyrtp_vad

```

```

set type RBDC

# Note: Even though "Static" is normally reserved for static
#       topologies, the satellite code will trigger a recalculation
#       of the routing tables whenever the topology changes.
#       Therefore, it is not so much "static" as "omniscient", in that
#       topology changes are known instantly throughout the topology.
#       See documentation for discussion of dynamic routing protocols.
$ns rtproto Static

#####
# Global configuration parameters
#####

global opt
set opt(chan)          Channel/Sat
set opt(bw_down)       8Mb; # Downlink bandwidth (satellite to ground)
set opt(bw_up)         2Mb; # Uplink bandwidth
set opt(phy)           Phy/Sat
set opt(mac)           Mac/Tdma
set opt(ifq)           Queue/DropTail
set opt(qlim)          1000
set opt(ll)            LL/Sat
set opt(wiredRouting) ON

# IMPORTANT This tracing enabling (trace-all) must precede link and node
#           creation. Then following all node, link, and error model
#           creation, invoke "$ns trace-all-satlinks $outfile"
set outfile [open $name.tr w]
$ns trace-all $outfile

#####
# Set up satellite and terrestrial nodes
#####

# GEO satellite: above Brazil -- lets put it at 61 deg. W (Satelite Amazonas)
$ns node-config -satNodeType geo-repeater \
    -phyType Phy/Repeater \
    -channelType $opt(chan) \
    -downlinkBW $opt(bw_down) \
    -wiredRouting $opt(wiredRouting)

set nodeSat [$ns node]
$nodeSat set-position -61

# Terminals: Let's put two within the US, two around the prime meridian
$ns node-config -satNodeType terminal
set nodeHub [$ns node]
set nodeTerm(1) [$ns node]
set nodeTerm(2) [$ns node]
set nodeTerm(3) [$ns node]
set nodeTerm(4) [$ns node]
set nodeTerm(5) [$ns node]
set nodeTerm(6) [$ns node]
set nodeTerm(7) [$ns node]
set nodeTerm(8) [$ns node]
set nodeTerm(9) [$ns node]
set nodeTerm(10) [$ns node]
set nodeTerm(11) [$ns node]
set nodeTerm(12) [$ns node]
set nodeTerm(13) [$ns node]
set nodeTerm(14) [$ns node]
set nodeTerm(15) [$ns node]
set nodeTerm(16) [$ns node]
set nodeTerm(17) [$ns node]
set nodeTerm(18) [$ns node]
set nodeTerm(19) [$ns node]
set nodeTerm(20) [$ns node]

$nodeHub set-position -22.9 -43.2; # Rio de Janeiro
$nodeTerm(1) set-position -1.5 -48.5 ;# Belém
$nodeTerm(2) set-position -19.9 -43.9 ;# Belo Horizonte
$nodeTerm(3) set-position 2.8 -60.7 ;# Boa Vista
$nodeTerm(4) set-position -15.8 -47.9 ;# Brasília
$nodeTerm(5) set-position -20.4 -54.6 ;# Campo Grande
$nodeTerm(6) set-position -15.6 -56.1 ;# Cuiabá
$nodeTerm(7) set-position -25.4 -49.3 ;# Curitiba
$nodeTerm(8) set-position -27.6 -48.5 ;# Florianópolis
$nodeTerm(9) set-position -3.7 -38.5 ;# Fortaleza
$nodeTerm(10) set-position 0.0 -51.1 ;# Macapá
$nodeTerm(11) set-position -9.7 -35.7 ;# Maceió
$nodeTerm(12) set-position -3.1 -60.0 ;# Manaus

```

```

$nodeTerm(13) set-position -30.0 -51.2;# Porto Alegre
$nodeTerm(14) set-position -8.8 -63.9 ;# Porto Velho
$nodeTerm(15) set-position -10.0 -67.8;# Rio Branco
$nodeTerm(16) set-position -13.0 -38.5;# Salvador
$nodeTerm(17) set-position -2.5 -44.3 ;# São Luís
$nodeTerm(18) set-position -23.5 -46.6;# São Paulo
$nodeTerm(19) set-position -5.1 -42.8 ;# Teresina
$nodeTerm(20) set-position -20.3 -40.3;# Vitória

#####
# Set up links
#####

# GSLS to the geo satellite:
$nodeHub add-gsl geo $opt(l1) $opt(ifq) $opt(qlim) $opt(mac) $opt(bw_up) \
    $opt(phy) [$nodeSat set downlink_] [$nodeSat set uplink_]
set req(0) [$nodeHub install-requester Requester/RBDC]
$nodeHub install-allocator Allocator/Proportional
$nodeHub insert-sat-ttl

for {set i 1} {$i <= 20} {incr i} {
    $nodeTerm($i) add-gsl geo $opt(l1) $opt(ifq) $opt(qlim) $opt(mac) $opt(bw_up) \
        $opt(phy) [$nodeSat set downlink_] [$nodeSat set uplink_]
    set req($i) [$nodeTerm($i) install-requester Requester/RBDC]
    $nodeTerm($i) insert-sat-ttl
}

#####
# Set up Core
#####

$ns node-config -reset
set nodeCore [$ns node]
set nodeMgw1 [$ns node]
set nodeMgw2 [$ns node]
set nodewww [$ns node]

$ns duplex-link $nodeHub $nodeCore 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodeMgw1 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodeMgw2 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodewww 50Mb 2ms DropTail

#####
# Set up Remote Nodes
#####
set initialnode 27
set numberVSATS 20 ;
for {set lannumber 1} {$lannumber <= $numberVSATS} {incr lannumber} {
    set lan($lannumber) [$ns newLan "" 0.5Mb 2ms LL Queue/DropTail MAC/802_3 Channel]
    for {set nodenumber [expr $initialnode + 5*($lannumber -1)]} {$nodenumber < [expr
$initialnode + 5*($lannumber -1)] +4} {incr nodenumber} {
        set n($nodenumber) [$ns node]
        $lan($lannumber) addNode "$n($nodenumber)" 0.5Mb 2ms
    }
    $lan($lannumber) addNode "$nodeTerm($lannumber)" 0.5Mb 2ms
}

#####
# Tracing
#####
$ns trace-all-satlinks $outfile

#####
# Set up RTP params
#####

set packetSize 94; # value in bytes
set ptime 0.04 ; # value in seconds
set RtpOn 360ms; # Mean time of ON periods
set RtpOff 650ms; # Mean time of OFF periods

#####
# Attach agents
# First Lan
#####

set lannumber 1

set nodenumber [expr $initialnode + 5*($lannumber -1)]

set s($nodenumber) [new Agent/UDP]
set Null($nodenumber) [new Agent/Null]
$ns attach-agent $n($nodenumber) $s($nodenumber)

```

```

$ns attach-agent $nodeMgw1 $Null($nodenumber)
$ns connect $s($nodenumber) $Null($nodenumber)
set Exp($nodenumber) [new Application/Traffic/Exponential]
$Exp($nodenumber) attach-agent $s($nodenumber)
$Exp($nodenumber) set packetSize_ $packetSize
$Exp($nodenumber) set burst_time_ $RtpOn
$Exp($nodenumber) set idle_time_ $RtpOff
$Exp($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
set f($nodenumber) [new Agent/UDP]
set Nullf($nodenumber) [new Agent/Null]
$ns attach-agent $nodeMgw1 $f($nodenumber)
$ns attach-agent $n($nodenumber) $Nullf($nodenumber)
$ns connect $f($nodenumber) $Nullf($nodenumber)
set Expf($nodenumber) [new Application/Traffic/Exponential]
$Expf($nodenumber) attach-agent $f($nodenumber)
$Expf($nodenumber) set packetSize_ $packetSize
$Expf($nodenumber) set burst_time_ $RtpOn
$Expf($nodenumber) set idle_time_ $RtpOff
$Expf($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
$ns at 4 "$Exp($nodenumber) start"
$ns at 4 "$Expf($nodenumber) start"
$ns at 60 "$Exp($nodenumber) stop"
$ns at 60 "$Expf($nodenumber) stop"

set secondnode [expr $nodenumber + 1]
set s($secondnode) [new Agent/UDP]
set Null($secondnode) [new Agent/Null]
$ns attach-agent $n($secondnode) $s($secondnode)
$ns attach-agent $nodeMgw2 $Null($secondnode)
$ns connect $s($secondnode) $Null($secondnode)
set Exp($secondnode) [new Application/Traffic/Exponential]
$Exp($secondnode) attach-agent $s($secondnode)
$Exp($secondnode) set packetSize_ $packetSize
$Exp($secondnode) set burst_time_ $RtpOn
$Exp($secondnode) set idle_time_ $RtpOff
$Exp($secondnode) set rate_ [expr 8*$packetSize/$ptime]
set f($secondnode) [new Agent/UDP]
set Nullf($secondnode) [new Agent/Null]
$ns attach-agent $nodeMgw2 $f($secondnode)
$ns attach-agent $n($secondnode) $Nullf($secondnode)
$ns connect $f($secondnode) $Nullf($secondnode)
set Expf($secondnode) [new Application/Traffic/Exponential]
$Expf($secondnode) attach-agent $f($secondnode)
$Expf($secondnode) set packetSize_ $packetSize
$Expf($secondnode) set burst_time_ $RtpOn
$Expf($secondnode) set idle_time_ $RtpOff
$Expf($secondnode) set rate_ [expr 8*$packetSize/$ptime]
$ns at 0 "$Exp($secondnode) start"
$ns at 0 "$Expf($secondnode) start"
$ns at 60 "$Exp($secondnode) stop"
$ns at 60 "$Expf($secondnode) stop"

#####
# Attach agents #
# Subsequent Lans #
#####

for {set lannumber 2} {$lannumber <= $numberVSATs} {incr lannumber} {

    set nodenumber [expr $initialnode + 5*($lannumber -1)]

    set s($nodenumber) [new Agent/UDP]
    set Null($nodenumber) [new Agent/Null]
    $ns attach-agent $n($nodenumber) $s($nodenumber)
    $ns attach-agent $nodeMgw2 $Null($nodenumber)
    $ns connect $s($nodenumber) $Null($nodenumber)
    set Exp($nodenumber) [new Application/Traffic/Exponential]
    $Exp($nodenumber) attach-agent $s($nodenumber)
    $Exp($nodenumber) set packetSize_ $packetSize
    $Exp($nodenumber) set burst_time_ $RtpOn
    $Exp($nodenumber) set idle_time_ $RtpOff
    $Exp($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
    set f($nodenumber) [new Agent/UDP]
    set Nullf($nodenumber) [new Agent/Null]
    $ns attach-agent $nodeMgw2 $f($nodenumber)
    $ns attach-agent $n($nodenumber) $Nullf($nodenumber)
    $ns connect $f($nodenumber) $Nullf($nodenumber)
    set Expf($nodenumber) [new Application/Traffic/Exponential]
    $Expf($nodenumber) attach-agent $f($nodenumber)
    $Expf($nodenumber) set packetSize_ $packetSize
    $Expf($nodenumber) set burst_time_ $RtpOn
    $Expf($nodenumber) set idle_time_ $RtpOff
    $Expf($nodenumber) set rate_ [expr 8*$packetSize/$ptime]

```

```

$ns at 0 "$Exp($nodenumber) start"
$ns at 0 "$Expf($nodenumber) start"
$ns at 60 "$Exp($nodenumber) stop"
$ns at 60 "$Expf($nodenumber) stop"

set secondnode [expr $nodenumber + 1]
set s($secondnode) [new Agent/UDP]
set Null($secondnode) [new Agent/Null]
$ns attach-agent $n($secondnode) $s($secondnode)
$ns attach-agent $nodeMgw2 $Null($secondnode)
$ns connect $s($secondnode) $Null($secondnode)
set Exp($secondnode) [new Application/Traffic/Exponential]
$Exp($secondnode) attach-agent $s($secondnode)
$Exp($secondnode) set packetSize_ $packetSize
$Exp($secondnode) set burst_time_ $RtpOn
$Exp($secondnode) set idle_time_ $RtpOff
$Exp($secondnode) set rate_ [expr 8*$packetSize/$ptime]
set f($secondnode) [new Agent/UDP]
set Nullf($secondnode) [new Agent/Null]
$ns attach-agent $nodeMgw2 $f($secondnode)
$ns attach-agent $n($secondnode) $Nullf($secondnode)
$ns connect $f($secondnode) $Nullf($secondnode)
set Expf($secondnode) [new Application/Traffic/Exponential]
$Expf($secondnode) attach-agent $f($secondnode)
$Expf($secondnode) set packetSize_ $packetSize
$Expf($secondnode) set burst_time_ $RtpOn
$Expf($secondnode) set idle_time_ $RtpOff
$Expf($secondnode) set rate_ [expr 8*$packetSize/$ptime]
$ns at "$Exp($secondnode) start"
$ns at "$Expf($secondnode) start"
$ns at 60 "$Exp($secondnode) stop"
$ns at 60 "$Expf($secondnode) stop"
}

#####
# Satellite routing #
#####

set satrouteobject_ [new SatRouteObject]
$satrouteobject_ compute_routes

$ns at 70 "finish"

proc finish {} {
    global ns outfile
    $ns flush-trace
    close $outfile
    $ns halt
}

$ns run

```

## C.4 Voz com VAD + Dados e Esquema Padrão

```

#####
#                                     #
#           variable Settings         #
#                                     #
#####

# Skyplex parameters
Mac/Tdma set max_slot_num_ 94          ;# slot per frame (preamble excluded)
Mac/Tdma set slot_packet_len_ 106      ;# slot size (bytes)
Mac/Tdma set num_frame_ 3              ;# multiframe size (in frames)
Mac/Tdma set bandwidth_ 2Mb
Allocator/Proportional set delay_ 1.4

set EPS 0.000001
set MAX_ACTIVE_NODE 64

LL/Sat set bandwidth_ 2Mb
Simulator instproc get-link {n1 n2} {
    set node1 [$self get-node-by-id $n1]
    return [$node1 set ll_(0)]
}
#

global ns

```

```

set ns [new Simulator]

set name satrtpt5_vad

set type RBDC

# Note: Even though "Static" is normally reserved for static
#       topologies, the satellite code will trigger a recalculation
#       of the routing tables whenever the topology changes.
#       Therefore, it is not so much "static" as "omniscient", in that
#       topology changes are known instantly throughout the topology.
#       See documentation for discussion of dynamic routing protocols.
$ns rtproto Static

#####
# Global configuration parameters
#####

global opt
set opt(chan)          Channel/Sat
set opt(bw_down)       8Mb; # Downlink bandwidth (satellite to ground)
set opt(bw_up)         2Mb; # Uplink bandwidth
set opt(phy)           Phy/Sat
set opt(mac)           Mac/Tdma
set opt(ifq)           Queue/DropTail
set opt(qlim)          1000
set opt(ll)            LL/Sat
set opt(wiredRouting) ON

# IMPORTANT This tracing enabling (trace-all) must precede link and node
#            creation. Then following all node, link, and error model
#            creation, invoke "$ns trace-all-satlinks $outfile"
set outfile [open $name.tr w]
$ns trace-all $outfile

set ev_file [open $name.out w]

#####
# Set up satellite and terrestrial nodes
#####

# GEO satellite: above Brazil -- lets put it at 61 deg. W (Satellite Amazonas)
$ns node-config -satNodeType geo-repeater \
    -phyType Phy/Repeater \
    -channelType $opt(chan) \
    -downlinkBw $opt(bw_down) \
    -wiredRouting $opt(wiredRouting)

set nodesSat [$ns node]
$nodesSat set-position -61

# Terminals: Let's put two within the US, two around the prime meridian
$ns node-config -satNodeType terminal
set nodeHub [$ns node]
set nodeTerm(1) [$ns node]
set nodeTerm(2) [$ns node]
set nodeTerm(3) [$ns node]
set nodeTerm(4) [$ns node]
set nodeTerm(5) [$ns node]
set nodeTerm(6) [$ns node]
set nodeTerm(7) [$ns node]
set nodeTerm(8) [$ns node]
set nodeTerm(9) [$ns node]
set nodeTerm(10) [$ns node]
set nodeTerm(11) [$ns node]
set nodeTerm(12) [$ns node]
set nodeTerm(13) [$ns node]
set nodeTerm(14) [$ns node]
set nodeTerm(15) [$ns node]
set nodeTerm(16) [$ns node]
set nodeTerm(17) [$ns node]
set nodeTerm(18) [$ns node]
set nodeTerm(19) [$ns node]
set nodeTerm(20) [$ns node]

$nodeHub set-position -22.9 -43.2; # Rio de Janeiro
$nodeTerm(1) set-position -1.5 -48.5 ;# Belém
$nodeTerm(2) set-position -19.9 -43.9 ;# Belo Horizonte
$nodeTerm(3) set-position 2.8 -60.7 ;# Boa Vista
$nodeTerm(4) set-position -15.8 -47.9 ;# Brasília
$nodeTerm(5) set-position -20.4 -54.6 ;# Campo Grande
$nodeTerm(6) set-position -15.6 -56.1 ;# Cuiabá
$nodeTerm(7) set-position -25.4 -49.3 ;# Curitiba

```

```

$nodeTerm(8) set-position -27.6 -48.5 ;# Florianópolis
$nodeTerm(9) set-position -3.7 -38.5 ;# Fortaleza
$nodeTerm(10) set-position 0.0 -51.1 ;# Macapá
$nodeTerm(11) set-position -9.7 -35.7 ;# Maceió
$nodeTerm(12) set-position -3.1 -60.0 ;# Manaus
$nodeTerm(13) set-position -30.0 -51.2 ;# Porto Alegre
$nodeTerm(14) set-position -8.8 -63.9 ;# Porto Velho
$nodeTerm(15) set-position -10.0 -67.8 ;# Rio Branco
$nodeTerm(16) set-position -13.0 -38.5 ;# Salvador
$nodeTerm(17) set-position -2.5 -44.3 ;# São Luís
$nodeTerm(18) set-position -23.5 -46.6 ;# São Paulo
$nodeTerm(19) set-position -5.1 -42.8 ;# Teresina
$nodeTerm(20) set-position -20.3 -40.3 ;# Vitória

#####
# Set up links
#####

# GSLS to the geo satellite:
$nodeHub add-gsl geo $opt(11) $opt(ifq) $opt(qlim) $opt(mac) $opt(bw_up) \
    $opt(phy) [$nodeSat set downlink_] [$nodeSat set uplink_]
set req(0) [$nodeHub install-requester Requester/RBDC]
$nodeHub install-allocator Allocator/Proportional
$nodeHub insert-sat-ttl
$nodeHub trace-event $ev_file

for {set i 1} {$i <= 20} {incr i} {
    $nodeTerm($i) add-gsl geo $opt(11) $opt(ifq) $opt(qlim) $opt(mac) $opt(bw_up) \
        $opt(phy) [$nodeSat set downlink_] [$nodeSat set uplink_]
    set req($i) [$nodeTerm($i) install-requester Requester/RBDC]
    $nodeTerm($i) insert-sat-ttl
}

#####
# Set up Core
#####

$ns node-config -reset
set nodeCore [$ns node]
set nodeMgw1 [$ns node]
set nodeMgw2 [$ns node]
set nodewww [$ns node]

$ns duplex-link $nodeHub $nodeCore 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodeMgw1 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodeMgw2 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodewww 50Mb 2ms DropTail

#####
# Set up Remote Nodes
#####
set initialnode 27
set numberVSATS 20 ;
for {set lannumber 1} {$lannumber <= $numberVSATS} {incr lannumber} {
    set lan($lannumber) [$ns newLan "" 0.5Mb 2ms LL Queue/DropTail MAC/802_3 Channel]
    for {set nodenumber [expr $initialnode + 5*($lannumber -1)]} {$nodenumber < [expr
$initialnode + 5*($lannumber -1)] +4} {incr nodenumber} {
        set n($nodenumber) [$ns node]
        $lan($lannumber) addNode "$n($nodenumber)" 0.5Mb 2ms
    }
    $lan($lannumber) addNode "$nodeTerm($lannumber)" 0.5Mb 2ms
}

#####
# Tracing
#####
$ns trace-all-satlinks $outfile
$nodeTerm(1) trace-event $ev_file

#####
# Set up HTTP
#####
# Create page pool as a central page generator. Use PagePool/Math
set pgp [new PagePool/Math]
set tmp [new RandomVariable/Constant] ;## Page size generator
$tmp set val_ 1024 ;## average page size
$pgp ranvar-size $tmp

set tmp [new RandomVariable/Exponential] ;## Page age generator
$tmp set avg_ 5 ;## average page age
$pgp ranvar-age $tmp

```

```

set server [new Http/Server $ns $nodewww] ;## Create a server and link it to the central page
pool
$server set-page-generator $pgp

set cache [new Http/Cache $ns $nodeCore] ;## Create a cache
$cache connect $server

#####
# Set up RTP params
#####

set packetSize 94; # Value in bytes
set ptime 0.04 ; # Value in seconds
set RtpOn 360ms; # Mean time of ON periods
set RtpOff 650ms; # Mean time of OFF periods

#####
# Attach agents
# First Lan
#####

set lannumber 1

set nodenumber [expr $initialnode + 5*($lannumber -1)]

set s($nodenumber) [new Agent/UDP]
set Null($nodenumber) [new Agent/Null]
$ns attach-agent $n($nodenumber) $s($nodenumber)
$ns attach-agent $nodeMgw1 $Null($nodenumber)
$ns connect $s($nodenumber) $Null($nodenumber)
set Exp($nodenumber) [new Application/Traffic/Exponential]
$Exp($nodenumber) attach-agent $s($nodenumber)
$Exp($nodenumber) set packetSize_ $packetSize
$Exp($nodenumber) set burst_time_ $RtpOn
$Exp($nodenumber) set idle_time_ $RtpOff
$Exp($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
set f($nodenumber) [new Agent/UDP]
set Nullf($nodenumber) [new Agent/Null]
$ns attach-agent $nodeMgw1 $f($nodenumber)
$ns attach-agent $n($nodenumber) $Nullf($nodenumber)
$ns connect $f($nodenumber) $Nullf($nodenumber)
set Expf($nodenumber) [new Application/Traffic/Exponential]
$Expf($nodenumber) attach-agent $f($nodenumber)
$Expf($nodenumber) set packetSize_ $packetSize
$Expf($nodenumber) set burst_time_ $RtpOn
$Expf($nodenumber) set idle_time_ $RtpOff
$Expf($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
$ns at 4 "$Exp($nodenumber) start"
$ns at 4 "$Expf($nodenumber) start"
$ns at 60 "$Exp($nodenumber) stop"
$ns at 60 "$Expf($nodenumber) stop"

set secondnode [expr $nodenumber + 1]
set s($secondnode) [new Agent/UDP]
set Null($secondnode) [new Agent/Null]
$ns attach-agent $n($secondnode) $s($secondnode)
$ns attach-agent $nodeMgw2 $Null($secondnode)
$ns connect $s($secondnode) $Null($secondnode)
set Exp($secondnode) [new Application/Traffic/Exponential]
$Exp($secondnode) attach-agent $s($secondnode)
$Exp($secondnode) set packetSize_ $packetSize
$Exp($secondnode) set burst_time_ $RtpOn
$Exp($secondnode) set idle_time_ $RtpOff
$Exp($secondnode) set rate_ [expr 8*$packetSize/$ptime]
set f($secondnode) [new Agent/UDP]
set Nullf($secondnode) [new Agent/Null]
$ns attach-agent $nodeMgw2 $f($secondnode)
$ns attach-agent $n($secondnode) $Nullf($secondnode)
$ns connect $f($secondnode) $Nullf($secondnode)
set Expf($secondnode) [new Application/Traffic/Exponential]
$Expf($secondnode) attach-agent $f($secondnode)
$Expf($secondnode) set packetSize_ $packetSize
$Expf($secondnode) set burst_time_ $RtpOn
$Expf($secondnode) set idle_time_ $RtpOff
$Expf($secondnode) set rate_ [expr 8*$packetSize/$ptime]
$ns at 0 "$Exp($secondnode) start"
$ns at 0 "$Expf($secondnode) start"
$ns at 60 "$Exp($secondnode) stop"
$ns at 60 "$Expf($secondnode) stop"

set thirdnode [expr $nodenumber + 2]
set client($lannumber) [new Http/Client $ns $n($thirdnode)] ;## Create a client
set tmp [new RandomVariable/Exponential] ;## Poisson process as request sequence

```



```

$tmp set avg_ 5 ;## average request interval
$client($lanumber) set-interval-generator $tmp
$client($lanumber) set-page-generator $pgp
$client($lanumber) connect $cache
$ns at 0.0 "$client($lanumber) start-session $cache $server"

set fourthnode [expr $nodenumber + 3]
set tcp($lanumber) [new Agent/TCP]
set tcpsink($lanumber) [new Agent/TCPSink]
$ns attach-agent $nodewww $tcp($lanumber)
$ns attach-agent $n($fourthnode) $tcpsink($lanumber)
$ns connect $tcp($lanumber) $tcpsink($lanumber)
set ftp($lanumber) [new Application/FTP]
$ftp($lanumber) attach-agent $tcp($lanumber)
$ns at 0 "$ftp($lanumber) start"
$ns at 60 "$ftp($lanumber) stop"

#####
# Attach agents
# Subsequent Lans
#####

for {set lanumber 2} {$lanumber <= $numberVSATs} {incr lanumber} {

    set nodenumber [expr $initialnode + 5*($lanumber -1)]

    set s($nodenumber) [new Agent/UDP]
    set Null($nodenumber) [new Agent/Null]
    $ns attach-agent $n($nodenumber) $s($nodenumber)
    $ns attach-agent $nodemgw2 $Null($nodenumber)
    $ns connect $s($nodenumber) $Null($nodenumber)
    set Exp($nodenumber) [new Application/Traffic/Exponential]
    $Exp($nodenumber) attach-agent $s($nodenumber)
    $Exp($nodenumber) set packetSize_ $packetSize
    $Exp($nodenumber) set burst_time_ $RtpOn
    $Exp($nodenumber) set idle_time_ $RtpOff
    $Exp($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
    set f($nodenumber) [new Agent/UDP]
    set Nullf($nodenumber) [new Agent/Null]
    $ns attach-agent $nodemgw2 $f($nodenumber)
    $ns attach-agent $n($nodenumber) $Nullf($nodenumber)
    $ns connect $f($nodenumber) $Nullf($nodenumber)
    set Expf($nodenumber) [new Application/Traffic/Exponential]
    $Expf($nodenumber) attach-agent $f($nodenumber)
    $Expf($nodenumber) set packetSize_ $packetSize
    $Expf($nodenumber) set burst_time_ $RtpOn
    $Expf($nodenumber) set idle_time_ $RtpOff
    $Expf($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
    $ns at 0 "$Exp($nodenumber) start"
    $ns at 0 "$Expf($nodenumber) start"
    $ns at 60 "$Exp($nodenumber) stop"
    $ns at 60 "$Expf($nodenumber) stop"

    set secondnode [expr $nodenumber + 1]
    set s($secondnode) [new Agent/UDP]
    set Null($secondnode) [new Agent/Null]
    $ns attach-agent $n($secondnode) $s($secondnode)
    $ns attach-agent $nodemgw2 $Null($secondnode)
    $ns connect $s($secondnode) $Null($secondnode)
    set Exp($secondnode) [new Application/Traffic/Exponential]
    $Exp($secondnode) attach-agent $s($secondnode)
    $Exp($secondnode) set packetSize_ $packetSize
    $Exp($secondnode) set burst_time_ $RtpOn
    $Exp($secondnode) set idle_time_ $RtpOff
    $Exp($secondnode) set rate_ [expr 8*$packetSize/$ptime]
    set f($secondnode) [new Agent/UDP]
    set Nullf($secondnode) [new Agent/Null]
    $ns attach-agent $nodemgw2 $f($secondnode)
    $ns attach-agent $n($secondnode) $Nullf($secondnode)
    $ns connect $f($secondnode) $Nullf($secondnode)
    set Expf($secondnode) [new Application/Traffic/Exponential]
    $Expf($secondnode) attach-agent $f($secondnode)
    $Expf($secondnode) set packetSize_ $packetSize
    $Expf($secondnode) set burst_time_ $RtpOn
    $Expf($secondnode) set idle_time_ $RtpOff
    $Expf($secondnode) set rate_ [expr 8*$packetSize/$ptime]
    $ns at 0 "$Exp($secondnode) start"
    $ns at 0 "$Expf($secondnode) start"
    $ns at 60 "$Exp($secondnode) stop"
    $ns at 60 "$Expf($secondnode) stop"

    set thirdnode [expr $nodenumber + 2]

```

```

set client($lanumber) [new Http/Client $ns $n($thirdnode)] ;## Create a client
set tmp [new RandomVariable/Exponential] ;## Poisson process as request sequence
$tmp set avg_ 5 ;## average request interval
$client($lanumber) set-interval-generator $tmp
$client($lanumber) set-page-generator $pgp
$client($lanumber) connect $cache
$ns at 0.0 "$client($lanumber) start-session $cache $server"

set fourthnode [expr $nodenumber + 3]
set tcp($lanumber) [new Agent/TCP]
set tcpsink($lanumber) [new Agent/TCPSink]
$ns attach-agent $nodewww $tcp($lanumber)
$ns attach-agent $n($fourthnode) $tcpsink($lanumber)
$ns connect $tcp($lanumber) $tcpsink($lanumber)
set ftp($lanumber) [new Application/FTP]
$ftp($lanumber) attach-agent $tcp($lanumber)
$ns at 0 "$ftp($lanumber) start"
$ns at 60 "$ftp($lanumber) stop"
}

#####
# Satellite routing
#####

set satrouteobject_ [new SatRouteObject]
$nsatrouteobject_ compute_routes

$ns at 70 "finish"

proc finish {} {
    global ns outfile
    $ns flush-trace
    close $outfile
    $ns halt
}

$ns run

```

## C.5 Voz com VAD + Dados e Pré-Alocação de Banda

```

#####
#
#           Allocator and Requester Definitions
#
#####

# initialize allocation variables
Allocator/Tcl instproc initvar { } {
    $self instvar active_node_ slot_index_ deficit bod_ fca_ ncra_
# Pre-Allocator_Support
    global active_voice_flow
# Pre-Allocator_Support

    set bod_ 1      ;# enable/disable BoD
    set fca_ 0      ;# enable/disable FCA (Free Capacity Assignment)

    set ncra_ 1     ;# number of slot per station allocated in CRA

    set slot_index_ 0

    for {set i 0} {$i < $active_node_} {incr i} {
        set deficit($i) 0.0
    }

# Pre-Allocator_Support
    for {set index 0} {$index < 22} {incr index} { ;# The first 21 nodes must be
satNodeType
        set active_voice_flow($i) 0
    }
# Pre-Allocator_Support
}

# allocation function
Allocator/Tcl instproc allocation { } {
    global EPS
    $self instvar active_node_ requestv_ max_slot_num_ schedule_
    $self instvar deficit slot_index_ bod_ fca_ ncra_

```

```

set totrreq 0
set slot 0
set ns 0
set si $slot_index_

for {set i 0} {$i < $active_node_} {incr i} {
    # clamp requestv_($i)
    if { $requestv_($i) > $max_slot_num_ } {
        set requestv_($i) $max_slot_num_
    }
    set totrreq [expr $totrreq + $requestv_($i)]
}

;# max number of slot allocated in BoD
set bodslot [expr $max_slot_num_-$active_node_*$ncra_]

if {$totrreq > $bodslot || ($fca_ == "1" && $totrreq > 0)} {
    for {set i 0} {$i < $active_node_} {incr i} {
        set deficit($i) [expr $deficit($i) + \
            double($bodslot)*$requestv_($i)/$totrreq]
    }
} else {
    for {set i 0} {$i < $active_node_} {incr i} {
        set deficit($i) [expr $deficit($i) + $requestv_($i)]
    }
}

;# Slot allocated in CRA
for {set i 0} {$i < $active_node_*$ncra_} {incr i} {
    set schedule_($i) [expr $i % $active_node_]
    incr slot
}

if { $bod_ == "1" } {
    ;# Slot allocated in BoD
    while { $slot < $max_slot_num_ && $ns < $active_node_ } {
        set ns 0

        while { $ns < $active_node_ && $deficit($slot_index_) <= $EPS } {
            set slot_index_ [expr ($slot_index_ + 1) % $active_node_]
            incr ns
        }

        if { $slot < $max_slot_num_ && $deficit($slot_index_) > $EPS } {
            set deficit($slot_index_) [expr $deficit($slot_index_)-1]
            set schedule_($slot) $slot_index_
            set slot_index_ [expr ($slot_index_ + 1) % $active_node_]
            incr slot
        }
    }
}

# remaining slot unallocated
if { $slot < $max_slot_num_ } {
    for {set i $slot} {$i < $max_slot_num_} {incr i} {
        set schedule_($i) -2
    }
}

}

Requester/Tcl instproc initvar { } {
    $self instvar alpha_ srate inp tin_
    $self instvar rbdc vbdc H_
    global type

    set H_ 1.0

    if { $type == "RBDC" } {
        set rbdc 1
        set vbdc 0
    }

    if { $type == "VBDC" } {
        set rbdc 0
        set vbdc 1
    }

    if { $type == "HYBR" } {
        set rbdc 1
        set vbdc 1
    }
}

```

```

        set alpha_ 0.5
        set srate 0

        set inp 0
        set tin_ 0
    }

    Requester/Tcl instproc dorequest { } {
        $self instvar request_ ifq_ qMonitor_ id_
        $self instvar alpha_ inp srate tin_ H_ node_
        $self instvar rbdc vbdc
    # Pre-Allocator_Support
        global active_voice_flow

        set NodeId $id_
    # Pre-Allocator_Support

        # Information related to node
        set mac [$node_ set mac_(0)]
        set Ls [$mac set slot_packet_len_]                ;# slot lenght
        set nf [$mac set num_frame_]                      ;# number of frames
        set Macband [Mac/Tdma set bandwidth_]
        set band [expr 10e6*[string trim $Macband Mb]]    ;# bandwidth (b/s)
        set Macslot [Mac/Tdma set max_slot_num_]
        set slots [expr $Macslot + 1]                    ;# slots
        set Df [expr $slots*8*$Ls/$band]                 ;# frame duration

        # get current time
        set ns_ [Simulator instance]
        set now [$ns_ now]

        # evaluate inputs
        set in [$qMonitor_ set barrivals_]
        set out [$qMonitor_ set bdepartures_]
        set drop [$qMonitor_ set bdrops_]

        set buffer [expr $in - $out - $drop]              ;# buffer sample
        set rate [expr $in - $inp]                        ;# rate sample

        if { $rate > $srate } {
            set srate $rate
        } else {
            set srate [expr $rate*$alpha_ + $srate*(1-$alpha_)] ;# smoothed rate
        }

    # Pre-Allocator_Support
        if { $active_voice_flow($NodeId) != 0 } {
            if { $active_voice_flow($NodeId) == 1 } {
                set voice_rate 283.974                    ;# (VoIP_rate/CRA_rate)*$nf*$Ls
            } else {
                if { $active_voice_flow($NodeId) == 2 } {
                    set voice_rate 567.95
                } else {
                    if { $active_voice_flow($NodeId) == 3 } {
                        set voice_rate 851.92
                    } else {
                        if { $active_voice_flow($NodeId) == 4 } {
                            set voice_rate 1135.9
                        } else {
                            if { $active_voice_flow($NodeId) == 5 } {
                                set voice_rate 1419.87
                            }
                        }
                    }
                }
            }
        }

        set srate $voice_rate
    }
    # Pre-Allocator_Support

    # the request is in slots per frame
    set nbuffer [expr ($buffer*$Df)/($Ls*$H_)]
    set nsrate [expr $srate/($Ls*$nf)]

    # calculate request
    set req [expr $nsrate*$rbdc + $nbuffer*$vbdc]

    # update counters
    set inp $in
    set tin_ $now

```

```

    return $req
}

#####
#                                     #
#       Variable Settings             #
#                                     #
#####

# Skyplex parameters
Mac/Tdma set max_slot_num_ 94        ;# slot per frame (preamble excluded)
Mac/Tdma set slot_packet_len_ 106     ;# slot size (bytes)
Mac/Tdma set num_frame_ 3             ;# multiframe size (in frames)
Mac/Tdma set bandwidth_ 2Mb
Allocator/Proportional set delay_ 1.4
Allocator/Tcl set delay_ 1.4

set EPS 0.000001
set MAX_ACTIVE_NODE 64

LL/Sat set bandwidth_ 2Mb

global ns
set ns [new Simulator]

set name satrt6_vad

set type RBDC

# Note: Even though "Static" is normally reserved for static
#       topologies, the satellite code will trigger a recalculation
#       of the routing tables whenever the topology changes.
#       Therefore, it is not so much "static" as "omniscient", in that
#       topology changes are known instantly throughout the topology.
#       See documentation for discussion of dynamic routing protocols.
$ns rtproto Static

#####
# Global configuration parameters      #
#####

global opt
set opt(chan) Channel/Sat
set opt(bw_down) 8Mb; # Downlink bandwidth (satellite to ground)
set opt(bw_up) 2Mb; # Uplink bandwidth
set opt(phy) Phy/Sat
set opt(mac) Mac/Tdma
set opt(ifq) Queue/DropTail
set opt(qlim) 1000
set opt(ll) LL/Sat
set opt(wiredRouting) ON

# IMPORTANT This tracing enabling (trace-all) must precede link and node
# creation. Then following all node, link, and error model
# creation, invoke "$ns trace-all-satlinks $outfile"
set outfile [open $name.tr w]
$ns trace-all $outfile

set ev_file [open $name.out w]

#####
# Set up satellite and terrestrial nodes
#####

# GEO satellite: above Brazil -- lets put it at 61 deg. W (Satellite Amazonas)
$ns node-config -satNodeType geo-repeater \
    -phyType Phy/Repeater \
    -channelType $opt(chan) \
    -downlinkBW $opt(bw_down) \
    -wiredRouting $opt(wiredRouting)

set nodesSat [$ns node]
$nodesSat set-position -61

# Terminals: Let's put two within the US, two around the prime meridian
$ns node-config -satNodeType terminal
set nodeHub [$ns node]
set nodeTerm(1) [$ns node]
set nodeTerm(2) [$ns node]
set nodeTerm(3) [$ns node]
set nodeTerm(4) [$ns node]
set nodeTerm(5) [$ns node]
set nodeTerm(6) [$ns node]

```

```

set nodeTerm(7) [$ns node]
set nodeTerm(8) [$ns node]
set nodeTerm(9) [$ns node]
set nodeTerm(10) [$ns node]
set nodeTerm(11) [$ns node]
set nodeTerm(12) [$ns node]
set nodeTerm(13) [$ns node]
set nodeTerm(14) [$ns node]
set nodeTerm(15) [$ns node]
set nodeTerm(16) [$ns node]
set nodeTerm(17) [$ns node]
set nodeTerm(18) [$ns node]
set nodeTerm(19) [$ns node]
set nodeTerm(20) [$ns node]

$nodeHub set-position -22.9 -43.2;      # Rio de Janeiro
$nodeTerm(1) set-position -1.5 -48.5 ;# Belém
$nodeTerm(2) set-position -19.9 -43.9 ;# Belo Horizonte
$nodeTerm(3) set-position 2.8 -60.7 ;# Boa Vista
$nodeTerm(4) set-position -15.8 -47.9 ;# Brasília
$nodeTerm(5) set-position -20.4 -54.6 ;# Campo Grande
$nodeTerm(6) set-position -15.6 -56.1 ;# Cuiabá
$nodeTerm(7) set-position -25.4 -49.3 ;# Curitiba
$nodeTerm(8) set-position -27.6 -48.5 ;# Florianópolis
$nodeTerm(9) set-position -3.7 -38.5 ;# Fortaleza
$nodeTerm(10) set-position 0.0 -51.1 ;# Macapá
$nodeTerm(11) set-position -9.7 -35.7 ;# Maceió
$nodeTerm(12) set-position -3.1 -60.0 ;# Manaus
$nodeTerm(13) set-position -30.0 -51.2 ;# Porto Alegre
$nodeTerm(14) set-position -8.8 -63.9 ;# Porto Velho
$nodeTerm(15) set-position -10.0 -67.8 ;# Rio Branco
$nodeTerm(16) set-position -13.0 -38.5 ;# Salvador
$nodeTerm(17) set-position -2.5 -44.3 ;# São Luís
$nodeTerm(18) set-position -23.5 -46.6 ;# São Paulo
$nodeTerm(19) set-position -5.1 -42.8 ;# Teresina
$nodeTerm(20) set-position -20.3 -40.3 ;# Vitória

#####
# Set up links
#####

# GSlS to the geo satellite:
$nodeHub add-gsl geo $opt(l1) $opt(ifq) $opt(qlim) $opt(mac) $opt(bw_up) \
    $opt(phy) [$nodeSat set downlink_] [$nodeSat set uplink_]
set req(0) [$nodeHub install-requester Requester/Tc1]
$nodeHub install-allocator Allocator/Tc1
$nodeHub insert-sat-ttl
$nodeHub trace-event $ev_file

for {set i 1} {$i <= 20} {incr i} {
    $nodeTerm($i) add-gsl geo $opt(l1) $opt(ifq) $opt(qlim) $opt(mac) $opt(bw_up) \
        $opt(phy) [$nodeSat set downlink_] [$nodeSat set uplink_]
    set req($i) [$nodeTerm($i) install-requester Requester/Tc1]
    $nodeTerm($i) insert-sat-ttl
}

#####
# Set up Core
#####

$ns node-config -reset
set nodeCore [$ns node]
set nodeMgw1 [$ns node]
set nodeMgw2 [$ns node]
set nodewww [$ns node]

$ns duplex-link $nodeHub $nodeCore 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodeMgw1 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodeMgw2 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodewww 50Mb 2ms DropTail

#####
# Set up Remote Nodes
#####
set initialnode 27
set numberVSATS 20 ;
for {set lannumber 1} {$lannumber <= $numberVSATS} {incr lannumber} {
    set lan($lannumber) [$ns newLan "" 0.5Mb 2ms LL Queue/DropTail MAC/802_3 Channel]
    for {set nodenumber [expr $initialnode + 5*($lannumber -1)]} {$nodenumber < [expr
$initialnode + 5*($lannumber -1)] +4} {incr nodenumber} {
        set n($nodenumber) [$ns node]
        $lan($lannumber) addNode "$n($nodenumber)" 0.5Mb 2ms
    }
}

```

```

    $lan($lannumber) addNode "$nodeTerm($lannumber)" 0.5Mb 2ms
}

#####
# Tracing #
#####
$ns trace-all-satlinks $outfile
$nodeTerm(1) trace-event $ev_file

#####
# Set up HTTP #
#####
# Create page pool as a central page generator. Use PagePool/Math
set pgp [new PagePool/Math]
set tmp [new RandomVariable/Constant] ;## Page size generator
$tmp set val_ 1024 ;## average page size
$pgp ranvar-size $tmp

set tmp [new RandomVariable/Exponential] ;## Page age generator
$tmp set avg_ 5 ;## average page age
$pgp ranvar-age $tmp

set server [new Http/Server $ns $nodeMgw] ;## Create a server and link it to the central page
pool
$server set-page-generator $pgp

set cache [new Http/Cache $ns $nodeCore] ;## Create a cache
$cache connect $server

#####
# Set up RTP params #
#####

set packetSize 94; # Value in bytes
set ptime 0.04 ; # Value in seconds
set RtpOn 360ms; # Mean time of ON periods
set RtpOff 650ms; # Mean time of OFF periods

#####
# Attach agents #
# First Lan #
#####

    set lannumber 1
# Pre-Allocator_Support
    set vSatIndex [expr $lannumber + 1] ;# id from node $nodeTerm(1)
# Pre-Allocator_Support

    set nodenumber [expr $initialnode + 5*($lannumber -1)]

    set s($nodenumber) [new Agent/UDP]
    set Null($nodenumber) [new Agent/Null]
    $ns attach-agent $n($nodenumber) $s($nodenumber)
    $ns attach-agent $nodeMgw1 $Null($nodenumber)
    $ns connect $s($nodenumber) $Null($nodenumber)
    set Exp($nodenumber) [new Application/Traffic/Exponential]
    $Exp($nodenumber) attach-agent $s($nodenumber)
    $Exp($nodenumber) set packetSize_ $packetSize
    $Exp($nodenumber) set burst_time_ $RtpOn
    $Exp($nodenumber) set idle_time_ $RtpOff
    $Exp($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
    set f($nodenumber) [new Agent/UDP]
    set Nullf($nodenumber) [new Agent/Null]
    $ns attach-agent $nodeMgw1 $f($nodenumber)
    $ns attach-agent $n($nodenumber) $Nullf($nodenumber)
    $ns connect $f($nodenumber) $Nullf($nodenumber)
    set Expf($nodenumber) [new Application/Traffic/Exponential]
    $Expf($nodenumber) attach-agent $f($nodenumber)
    $Expf($nodenumber) set packetSize_ $packetSize
    $Expf($nodenumber) set burst_time_ $RtpOn
    $Expf($nodenumber) set idle_time_ $RtpOff
    $Expf($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
    $ns at 4 "$Exp($nodenumber) start"
    $ns at 4 "$Expf($nodenumber) start"
# Pre-Allocator_Support
    $ns at 4 "set active_voice_flow($vSatIndex) 2"
# Pre-Allocator_Support
    $ns at 60 "$Exp($nodenumber) stop"
    $ns at 60 "$Expf($nodenumber) stop"
# Pre-Allocator_Support
    $ns at 60 "set active_voice_flow($vSatIndex) 1"
# Pre-Allocator_Support

```

```

set secondnode [expr $nodenumber + 1]
set s($secondnode) [new Agent/UDP]
set Null($secondnode) [new Agent/Null]
$ns attach-agent $n($secondnode) $s($secondnode)
$ns attach-agent $nodeMgw2 $Null($secondnode)
$ns connect $s($secondnode) $Null($secondnode)
set Exp($secondnode) [new Application/Traffic/Exponential]
$Exp($secondnode) attach-agent $s($secondnode)
$Exp($secondnode) set packetSize_ $packetSize
$Exp($secondnode) set burst_time_ $RtpOn
$Exp($secondnode) set idle_time_ $RtpOff
$Exp($secondnode) set rate_ [expr 8*$packetSize/$ptime]
set f($secondnode) [new Agent/UDP]
set Nullf($secondnode) [new Agent/Null]
$ns attach-agent $nodeMgw2 $f($secondnode)
$ns attach-agent $n($secondnode) $Nullf($secondnode)
$ns connect $f($secondnode) $Nullf($secondnode)
set Expf($secondnode) [new Application/Traffic/Exponential]
$Expf($secondnode) attach-agent $f($secondnode)
$Expf($secondnode) set packetSize_ $packetSize
$Expf($secondnode) set burst_time_ $RtpOn
$Expf($secondnode) set idle_time_ $RtpOff
$Expf($secondnode) set rate_ [expr 8*$packetSize/$ptime]
$ns at [expr 0.3 + ($lanumber - 1)] "$Exp($secondnode) start"
$ns at [expr 0.3 + ($lanumber - 1)] "$Expf($secondnode) start"
# Pre-Allocator_Support
$ns at [expr 0.3 + ($lanumber - 1)] "set active_voice_flow($VSatIndex) 1"
# Pre-Allocator_Support
$ns at 60 "$Exp($secondnode) stop"
$ns at 60 "$Expf($secondnode) stop"
# Pre-Allocator_Support
$ns at 60.02 "set active_voice_flow($VSatIndex) 0"
# Pre-Allocator_Support

set thirdnode [expr $nodenumber + 2]
set client($lanumber) [new Http/Client $ns $n($thirdnode)] ;## Create a client
set tmp [new RandomVariable/Exponential] ;## Poisson process as request sequence
$tmp set avg_ 5 ;## average request interval
$client($lanumber) set-interval-generator $tmp
$client($lanumber) set-page-generator $pgp
$client($lanumber) connect $cache
$ns at 0.0 "$client($lanumber) start-session $cache $server"

set fourthnode [expr $nodenumber + 3]
set tcp($lanumber) [new Agent/TCP]
set tcpsink($lanumber) [new Agent/TCPSink]
$ns attach-agent $nodewww $tcp($lanumber)
$ns attach-agent $n($fourthnode) $tcpsink($lanumber)
$ns connect $tcp($lanumber) $tcpsink($lanumber)
set ftp($lanumber) [new Application/FTP]
$ftp($lanumber) attach-agent $tcp($lanumber)
$ns at 0 "$ftp($lanumber) start"
$ns at 60 "$ftp($lanumber) stop"

#####
# Attach agents
# Subsequent Lans
#####

for {set lanumber 2} {$lanumber <= $numberVSats} {incr lanumber} {

# Pre-Allocator_Support
set VSatIndex [expr $lanumber + 1]
# Pre-Allocator_Support
set nodenumber [expr $initialnode + 5*($lanumber -1)]

set s($nodenumber) [new Agent/UDP]
set Null($nodenumber) [new Agent/Null]
$ns attach-agent $n($nodenumber) $s($nodenumber)
$ns attach-agent $nodeMgw2 $Null($nodenumber)
$ns connect $s($nodenumber) $Null($nodenumber)
set Exp($nodenumber) [new Application/Traffic/Exponential]
$Exp($nodenumber) attach-agent $s($nodenumber)
$Exp($nodenumber) set packetSize_ $packetSize
$Exp($nodenumber) set burst_time_ $RtpOn
$Exp($nodenumber) set idle_time_ $RtpOff
$Exp($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
set f($nodenumber) [new Agent/UDP]
set Nullf($nodenumber) [new Agent/Null]
$ns attach-agent $nodeMgw2 $f($nodenumber)
$ns attach-agent $n($nodenumber) $Nullf($nodenumber)
$ns connect $f($nodenumber) $Nullf($nodenumber)

```



```

    set Expf($nodenumber) [new Application/Traffic/Exponential]
    $Expf($nodenumber) attach-agent $f($nodenumber)
    $Expf($nodenumber) set packetSize_ $packetSize
    $Expf($nodenumber) set burst_time_ $RtpOn
    $Expf($nodenumber) set idle_time_ $RtpOff
    $Expf($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
    $ns at [expr 0.1 + ($lanumber - 1)] "$Exp($nodenumber) start"
    $ns at [expr 0.1 + ($lanumber - 1)] "$Expf($nodenumber) start"
# Pre-Allocator_Support
    $ns at [expr 0.1 + ($lanumber - 1)] "set active_voice_flow($VSatIndex) 1"
# Pre-Allocator_Support
    $ns at 60 "$Exp($nodenumber) stop"
    $ns at 60 "$Expf($nodenumber) stop"
# Pre-Allocator_Support
    $ns at 60 "set active_voice_flow($VSatIndex) 1"
# Pre-Allocator_Support

    set secondnode [expr $nodenumber + 1]
    set s($secondnode) [new Agent/UDP]
    set Null($secondnode) [new Agent/Null]
    $ns attach-agent $n($secondnode) $s($secondnode)
    $ns attach-agent $nodeMgw2 $Null($secondnode)
    $ns connect $s($secondnode) $Null($secondnode)
    set Exp($secondnode) [new Application/Traffic/Exponential]
    $Exp($secondnode) attach-agent $s($secondnode)
    $Exp($secondnode) set packetSize_ $packetSize
    $Exp($secondnode) set burst_time_ $RtpOn
    $Exp($secondnode) set idle_time_ $RtpOff
    $Exp($secondnode) set rate_ [expr 8*$packetSize/$ptime]
    set f($secondnode) [new Agent/UDP]
    set Nullf($secondnode) [new Agent/Null]
    $ns attach-agent $nodeMgw2 $f($secondnode)
    $ns attach-agent $n($secondnode) $Nullf($secondnode)
    $ns connect $f($secondnode) $Nullf($secondnode)
    set Expf($secondnode) [new Application/Traffic/Exponential]
    $Expf($secondnode) attach-agent $f($secondnode)
    $Expf($secondnode) set packetSize_ $packetSize
    $Expf($secondnode) set burst_time_ $RtpOn
    $Expf($secondnode) set idle_time_ $RtpOff
    $Expf($secondnode) set rate_ [expr 8*$packetSize/$ptime]
    $ns at [expr 0.3 + ($lanumber - 1)] "$Exp($secondnode) start"
    $ns at [expr 0.3 + ($lanumber - 1)] "$Expf($secondnode) start"
# Pre-Allocator_Support
    $ns at [expr 0.3 + ($lanumber - 1)] "set active_voice_flow($VSatIndex) 2"
# Pre-Allocator_Support
    $ns at 60 "$Exp($secondnode) stop"
    $ns at 60 "$Expf($secondnode) stop"
# Pre-Allocator_Support
    $ns at 60.02 "set active_voice_flow($VSatIndex) 0"
# Pre-Allocator_Support

    set thirdnode [expr $nodenumber + 2]
    set client($lanumber) [new Http/Client $ns $n($thirdnode)] ;## Create a client
    set tmp [new RandomVariable/Exponential] ;## Poisson process as request sequence
    $tmp set avg_ 5 ;## average request interval
    $client($lanumber) set-interval-generator $tmp
    $client($lanumber) set-page-generator $pgp
    $client($lanumber) connect $cache
    $ns at 0.0 "$client($lanumber) start-session $cache $server"

    set fourthnode [expr $nodenumber + 3]
    set tcp($lanumber) [new Agent/TCP]
    set tcpsink($lanumber) [new Agent/TCPSink]
    $ns attach-agent $nodewww $tcp($lanumber)
    $ns attach-agent $n($fourthnode) $tcpsink($lanumber)
    $ns connect $tcp($lanumber) $tcpsink($lanumber)
    set ftp($lanumber) [new Application/FTP]
    $ftp($lanumber) attach-agent $tcp($lanumber)
    $ns at 0 "$ftp($lanumber) start"
    $ns at 60 "$ftp($lanumber) stop"
}

#####
# Satellite routing
#####

set satrouteobject_ [new SatRouteObject]
$satrouteobject_ compute_routes

$ns at 70 "finish"

proc finish {} {
    global ns outfile

```

```

        $ns flush-trace
        close $outfile
        $ns halt
    }

$ns run

```

## C.6 Voz com VAD + Dados e Alocação Padrão em alto tráfego

```

#####
#                                     #
#       variable Settings              #
#                                     #
#####

# Skyplex parameters
Mac/Tdma set max_slot_num_ 94          ;# slot per frame (preamble excluded)
Mac/Tdma set slot_packet_len_ 106      ;# slot size (bytes)
Mac/Tdma set num_frame_ 3              ;# multiframe size (in frames)
Mac/Tdma set bandwidth_ 2Mb
Allocator/Proportional set delay_ 1.4
Allocator/Tcl set delay_ 1.4

set EPS 0.000001
set MAX_ACTIVE_NODE 64

LL/Sat set bandwidth_ 2Mb
Simulator instproc get-link {n1 n2} {
    set node1 [$self get-node-by-id $n1]
    return [$node1 set ll_(0)]
}
#

global ns
set ns [new Simulator]

set name satrtpt7_vad

set type RBDC

# Note: Even though "Static" is normally reserved for static
#        topologies, the satellite code will trigger a recalculation
#        of the routing tables whenever the topology changes.
#        Therefore, it is not so much "static" as "omniscient", in that
#        topology changes are known instantly throughout the topology.
#        See documentation for discussion of dynamic routing protocols.
$ns rtproto Static

#####
# Global configuration parameters      #
#####

global opt
set opt(chan) Channel/Sat
set opt(bw_down) 8Mb; # Downlink bandwidth (satellite to ground)
set opt(bw_up) 2Mb; # Uplink bandwidth
set opt(phy) Phy/Sat
set opt(mac) Mac/Tdma
set opt(ifq) Queue/DropTail
set opt(qlim) 1000
set opt(ll) LL/Sat
set opt(wiredRouting) ON

# IMPORTANT This tracing enabling (trace-all) must precede link and node
# creation. Then following all node, link, and error model
# creation, invoke "$ns trace-all-satlinks $outfile"
set outfile [open $name.tr w]
$ns trace-all $outfile

set ev_file [open $name.out w]

#####
# Set up satellite and terrestrial nodes
#####

# GEO satellite: above Brazil -- lets put it at 61 deg. w (Satelite Amazonas)
$ns node-config -satNodeType geo-repeater \
    -phyType Phy/Repeater \

```

```

        -channelType $opt(chan) \
        -downlinkBW $opt(bw_down) \
        -wiredRouting $opt(wiredRouting)

set nodeSat [$ns node]
$nodeSat set-position -61

# Terminals:
$ns node-config -satNodeType terminal
set nodeHub [$ns node]
set nodeTerm(1) [$ns node]
set nodeTerm(2) [$ns node]

$nodeHub set-position -22.9 -43.2;      # Rio de Janeiro
$nodeTerm(1) set-position -1.5 -48.5 ;# Belém
$nodeTerm(2) set-position -19.9 -43.9 ;# Belo Horizonte

#####
# Set up links
#####

# GSLS to the geo satellite:
$nodeHub add-gsl geo $opt(ll) $opt(iff) $opt(qlim) $opt(mac) $opt(bw_up) \
    $opt(phy) [$nodeSat set downlink_] [$nodeSat set uplink_]
set req(0) [$nodeHub install-requester Requester/RBDC]
$nodeHub install-allocator Allocator/Proportional
$nodeHub insert-sat-ttl
$nodeHub trace-event $ev_file

for {set i 1} {$i <= 2} {incr i} {
    $nodeTerm($i) add-gsl geo $opt(ll) $opt(iff) $opt(qlim) $opt(mac) $opt(bw_up) \
        $opt(phy) [$nodeSat set downlink_] [$nodeSat set uplink_]
    set req($i) [$nodeTerm($i) install-requester Requester/RBDC]
    $nodeTerm($i) insert-sat-ttl
}

#####
# Set up Core
#####

$ns node-config -reset
set nodeCore [$ns node]
set nodeMgw1 [$ns node]
set nodeMgw2 [$ns node]
set nodewww [$ns node]

$ns duplex-link $nodeHub $nodeCore 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodeMgw1 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodeMgw2 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodewww 50Mb 2ms DropTail

#####
# Set up Remote Nodes
#####
set initialnode 9
set numberVSATS 2 ;
for {set lannumber 1} {$lannumber <= $numberVSATS} {incr lannumber} {
    set lan($lannumber) [$ns newLan "" 0.5Mb 2ms LL Queue/DropTail MAC/802_3 Channel]
    for {set nodenumber [expr $initialnode + 23*($lannumber -1)]} {$nodenumber < [expr
$initialnode + 23*($lannumber -1)] + 22} {incr nodenumber} {
        set n($nodenumber) [$ns node]
        $lan($lannumber) addNode "$n($nodenumber)" 0.5Mb 2ms
    }
    $lan($lannumber) addNode "$nodeTerm($lannumber)" 0.5Mb 2ms
}

#####
# Tracing
#####
$ns trace-all-satlinks $outfile
$nodeTerm(1) trace-event $ev_file

#####
# Set up HTTP
#####
# Create page pool as a central page generator. Use PagePool/Math
set pgp [new PagePool/Math]
set tmp [new RandomVariable/Constant] ;## Page size generator
$tmp set val_ 1024 ;## average page size
$pgp ranvar-size $tmp

set tmp [new RandomVariable/Exponential] ;## Page age generator
$tmp set avg_ 5 ;## average page age

```

```

$pgp ranvar-age $tmp

set server [new Http/Server $ns $nodewww] ;## Create a server and link it to the central page
pool
$server set-page-generator $pgp

set cache [new Http/Cache $ns $nodeCore] ;## Create a cache
$cache connect $server

#####
# Set up RTP params
#####

set packetSize 94; # Value in bytes
set ptime 0.04 ; # Value in seconds
set RtpOn 360ms; # Mean time of ON periods
set RtpOff 650ms; # Mean time of OFF periods

#####
# Attach agents
# First Lan
#####

set lannumber 1

set nodenumber [expr $initialnode + 23*($lannumber -1)]

set s($nodenumber) [new Agent/UDP]
set Null($nodenumber) [new Agent/Null]
$ns attach-agent $n($nodenumber) $s($nodenumber)
$ns attach-agent $nodeMgw1 $Null($nodenumber)
$ns connect $s($nodenumber) $Null($nodenumber)
set Exp($nodenumber) [new Application/Traffic/Exponential]
$Exp($nodenumber) attach-agent $s($nodenumber)
$Exp($nodenumber) set packetSize_ $packetSize
$Exp($nodenumber) set burst_time_ $RtpOn
$Exp($nodenumber) set idle_time_ $RtpOff
$Exp($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
set f($nodenumber) [new Agent/UDP]
set Nullf($nodenumber) [new Agent/Null]
$ns attach-agent $nodeMgw1 $f($nodenumber)
$ns attach-agent $n($nodenumber) $Nullf($nodenumber)
$ns connect $f($nodenumber) $Nullf($nodenumber)
set Expf($nodenumber) [new Application/Traffic/Exponential]
$Expf($nodenumber) attach-agent $f($nodenumber)
$Expf($nodenumber) set packetSize_ $packetSize
$Expf($nodenumber) set burst_time_ $RtpOn
$Expf($nodenumber) set idle_time_ $RtpOff
$Expf($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
$ns at 4 "$Exp($nodenumber) start"
$ns at 4 "$Expf($nodenumber) start"
$ns at 60 "$Exp($nodenumber) stop"
$ns at 60 "$Expf($nodenumber) stop"

for {set nodecounter 1} {$nodecounter <= 19} {incr nodecounter} {

set secondnode [expr $nodenumber + $nodecounter]
set s($secondnode) [new Agent/UDP]
set Null($secondnode) [new Agent/Null]
$ns attach-agent $n($secondnode) $s($secondnode)
$ns attach-agent $nodeMgw2 $Null($secondnode)
$ns connect $s($secondnode) $Null($secondnode)
set Exp($secondnode) [new Application/Traffic/Exponential]
$Exp($secondnode) attach-agent $s($secondnode)
$Exp($secondnode) set packetSize_ $packetSize
$Exp($secondnode) set burst_time_ $RtpOn
$Exp($secondnode) set idle_time_ $RtpOff
$Exp($secondnode) set rate_ [expr 8*$packetSize/$ptime]
set f($secondnode) [new Agent/UDP]
set Nullf($secondnode) [new Agent/Null]
$ns attach-agent $nodeMgw2 $f($secondnode)
$ns attach-agent $n($secondnode) $Nullf($secondnode)
$ns connect $f($secondnode) $Nullf($secondnode)
set Expf($secondnode) [new Application/Traffic/Exponential]
$Expf($secondnode) attach-agent $f($secondnode)
$Expf($secondnode) set packetSize_ $packetSize
$Expf($secondnode) set burst_time_ $RtpOn
$Expf($secondnode) set idle_time_ $RtpOff
$Expf($secondnode) set rate_ [expr 8*$packetSize/$ptime]
$ns at [expr 0.3 + ($nodecounter - 1)] "$Exp($secondnode) start"
$ns at [expr 0.3 + ($nodecounter - 1)] "$Expf($secondnode) start"
$ns at 60 "$Exp($secondnode) stop"
$ns at 60 "$Expf($secondnode) stop"
}

```

```

}

set thirdnode [expr $nodenumber + 20]
set client($lannumber) [new Http/Client $ns $n($thirdnode)] ;## Create a client
set tmp [new RandomVariable/Exponential] ;## Poisson process as request sequence
$tmp set avg_ 5 ;## average request interval
$client($lannumber) set-interval-generator $tmp
$client($lannumber) set-page-generator $pgp
$client($lannumber) connect $cache
$ns at 0.0 "$client($lannumber) start-session $cache $server"

set fourthnode [expr $nodenumber + 21]
set tcp($lannumber) [new Agent/TCP]
set tcpsink($lannumber) [new Agent/TCPSink]
$ns attach-agent $nodewww $tcp($lannumber)
$ns attach-agent $n($fourthnode) $tcpsink($lannumber)
$ns connect $tcp($lannumber) $tcpsink($lannumber)
set ftp($lannumber) [new Application/FTP]
$ftp($lannumber) attach-agent $tcp($lannumber)
$ns at 0 "$ftp($lannumber) start"
$ns at 60 "$ftp($lannumber) stop"

#####
# Attach agents                                     #
# Subsequent Lans                                   #
#####

for {set lannumber 2} {$lannumber <= $numberVSATs} {incr lannumber} {

    set nodenumber [expr $initialnode + 23*($lannumber - 1)]

    set s($nodenumber) [new Agent/UDP]
    set Null($nodenumber) [new Agent/Null]
    $ns attach-agent $n($nodenumber) $s($nodenumber)
    $ns attach-agent $nodemgw2 $Null($nodenumber)
    $ns connect $s($nodenumber) $Null($nodenumber)
    set Exp($nodenumber) [new Application/Traffic/Exponential]
    $Exp($nodenumber) attach-agent $s($nodenumber)
    $Exp($nodenumber) set packetSize_ $packetSize
    $Exp($nodenumber) set burst_time_ $RtpOn
    $Exp($nodenumber) set idle_time_ $RtpOff
    $Exp($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
    set f($nodenumber) [new Agent/UDP]
    set Nullf($nodenumber) [new Agent/Null]
    $ns attach-agent $nodemgw2 $f($nodenumber)
    $ns attach-agent $n($nodenumber) $Nullf($nodenumber)
    $ns connect $f($nodenumber) $Nullf($nodenumber)
    set Expf($nodenumber) [new Application/Traffic/Exponential]
    $Expf($nodenumber) attach-agent $f($nodenumber)
    $Expf($nodenumber) set packetSize_ $packetSize
    $Expf($nodenumber) set burst_time_ $RtpOn
    $Expf($nodenumber) set idle_time_ $RtpOff
    $Expf($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
    $ns at [expr 0.1 + ($lannumber - 1)] "$Exp($nodenumber) start"
    $ns at [expr 0.1 + ($lannumber - 1)] "$Expf($nodenumber) start"
    $ns at 60 "$Exp($nodenumber) stop"
    $ns at 60 "$Expf($nodenumber) stop"

    for {set nodecounter 1} {$nodecounter <= 19} {incr nodecounter} {

        set secondnode [expr $nodenumber + 1]
        set s($secondnode) [new Agent/UDP]
        set Null($secondnode) [new Agent/Null]
        $ns attach-agent $n($secondnode) $s($secondnode)
        $ns attach-agent $nodemgw2 $Null($secondnode)
        $ns connect $s($secondnode) $Null($secondnode)
        set Exp($secondnode) [new Application/Traffic/Exponential]
        $Exp($secondnode) attach-agent $s($secondnode)
        $Exp($secondnode) set packetSize_ $packetSize
        $Exp($secondnode) set burst_time_ $RtpOn
        $Exp($secondnode) set idle_time_ $RtpOff
        $Exp($secondnode) set rate_ [expr 8*$packetSize/$ptime]
        set f($secondnode) [new Agent/UDP]
        set Nullf($secondnode) [new Agent/Null]
        $ns attach-agent $nodemgw2 $f($secondnode)
        $ns attach-agent $n($secondnode) $Nullf($secondnode)
        $ns connect $f($secondnode) $Nullf($secondnode)
        set Expf($secondnode) [new Application/Traffic/Exponential]
        $Expf($secondnode) attach-agent $f($secondnode)
        $Expf($secondnode) set packetSize_ $packetSize
        $Expf($secondnode) set burst_time_ $RtpOn
    }
}

```

```

    $Expf($secondnode) set idle_time_ $Rtpoff
    $Expf($secondnode) set rate_ [expr 8*$packetSize/$ptime]
    $ns at [expr 0.1 + ($nodecounter - 1)] "$Exp($secondnode) start"
    $ns at [expr 0.1 + ($nodecounter - 1)] "$Expf($secondnode) start"
    $ns at 60 "$Exp($secondnode) stop"
    $ns at 60 "$Expf($secondnode) stop"
}

set thirdnode [expr $nodenumber + 20]
set client($lannumber) [new Http/Client $ns $n($thirdnode)] ;## Create a client
set tmp [new RandomVariable/Exponential] ;## Poisson process as request sequence
$tmp set avg_ 5 ;## average request interval
$client($lannumber) set-interval-generator $tmp
$client($lannumber) set-page-generator $pgp
$client($lannumber) connect $cache
$ns at 0.0 "$client($lannumber) start-session $cache $server"

set fourthnode [expr $nodenumber + 21]
set tcp($lannumber) [new Agent/TCP]
set tcpsink($lannumber) [new Agent/TCPSink]
$ns attach-agent $nodewww $tcp($lannumber)
$ns attach-agent $n($fourthnode) $tcpsink($lannumber)
$ns connect $tcp($lannumber) $tcpsink($lannumber)
set ftp($lannumber) [new Application/FTP]
$ftp($lannumber) attach-agent $tcp($lannumber)
$ns at 0 "$ftp($lannumber) start"
$ns at 60 "$ftp($lannumber) stop"
}

#####
# Satellite routing                                     #
#####

set satrouteobject_ [new SatRouteObject]
$satrouteobject_ compute_routes

$ns at 70 "finish"

proc finish {} {
    global ns outfile
    $ns flush-trace
    close $outfile
    $ns halt
}

$ns run

```

## C.7 Voz com VAD + Dados com Pré-Alocação em alto tráfego

```

#####
#                                     #
#       Allocator and Requester Definitions                               #
#                                     #
#####

# initialize allocation variables
Allocator/Tcl instproc initvar { } {
    $self instvar active_node_ slot_index_ deficit bod_ fca_ ncra_

# Pre-Allocator_Support
    global active_voice_flow
# Pre-Allocator_Support

    set bod_ 1      ;# enable/disable BoD
    set fca_ 0      ;# enable/disable FCA (Free Capacity Assignment)

    set ncra_ 1     ;# number of slot per station allocated in CRA

    set slot_index_ 0

    for {set i 0} {$i < $active_node_} {incr i} {
        set deficit($i) 0.0
    }

# Pre-Allocator_Support
    for {set index 0} {$index < 4} {incr index} { ;# The first 3 nodes must be satNodeType
        set active_voice_flow($i) 0
    }
}

```

```

    }
# Pre-Allocator_Support
}

# allocation function
Allocator/Tcl instproc allocation { } {
    global EPS
    $self instvar active_node_ requestv_ max_slot_num_ schedule_
    $self instvar deficit slot_index_ bod_ fca_ ncra_

    set totreq 0
    set slot 0
    set ns 0
    set si $slot_index_

    for {set i 0} {$i < $active_node_} {incr i} {
        # clamp requestv_($i)
        if { $requestv_($i) > $max_slot_num_ } {
            set requestv_($i) $max_slot_num_
        }
        set totreq [expr $totreq + $requestv_($i)]
    }

    ;# max number of slot allocated in BoD
    set bodslot [expr $max_slot_num_-$active_node_*$ncra_]

    if { $totreq > $bodslot || ($fca_ == "1" && $totreq > 0) } {
        for {set i 0} {$i < $active_node_} {incr i} {
            set deficit($i) [expr $deficit($i) + \
                double($bodslot)*$requestv_($i)/$totreq]
        }
    } else {
        for {set i 0} {$i < $active_node_} {incr i} {
            set deficit($i) [expr $deficit($i) + $requestv_($i)]
        }
    }

    ;# Slot allocated in CRA
    for {set i 0} {$i < $active_node_*$ncra_} {incr i} {
        set schedule_($i) [expr $i % $active_node_]
        incr slot
    }

    if { $bod_ == "1" } {
        ;# Slot allocated in BoD
        while { $slot < $max_slot_num_ && $ns < $active_node_ } {
            set ns 0

            while { $ns < $active_node_ && $deficit($slot_index_) <= $EPS } {
                set slot_index_ [expr ($slot_index_ + 1) % $active_node_]
                incr ns
            }

            if { $slot < $max_slot_num_ && $deficit($slot_index_) > $EPS } {
                set deficit($slot_index_) [expr $deficit($slot_index_)-1]
                set schedule_($slot) $slot_index_
                set slot_index_ [expr ($slot_index_ + 1) % $active_node_]
                incr slot
            }
        }
    }

    ;# remaining slot unallocated
    if { $slot < $max_slot_num_ } {
        for {set i $slot} {$i < $max_slot_num_} {incr i} {
            set schedule_($i) -2
        }
    }
}

Requester/Tcl instproc initvar { } {
    $self instvar alpha_ srate inp tin_
    $self instvar rbdc vbdc H_
    global type

    set H_ 1.0

    if { $type == "RBDC" } {
        set rbdc 1
        set vbdc 0
    }
}

```

```

    if { $type == "VBDC" } {
        set rbdc 0
        set vbdc 1
    }

    if { $type == "HYBR" } {
        set rbdc 1
        set vbdc 1
    }

    set alpha_ 0.5
    set srate 0

    set inp 0
    set tin_ 0
}

Requester/Tcl instproc dorequest { } {
    $self instvar request_ ifq_ qMonitor_ id_
    $self instvar alpha_ inp srate tin_ H_ node_
    $self instvar rbdc vbdc
# Pre-Allocator_Support
    global active_voice_flow

    set NodeId $id_
# Pre-Allocator_Support

    # Information related to node
    set mac [$node_ set mac_(0)]
    set Ls [$mac set slot_packet_len_]           ;# slot length
    set nf [$mac set num_frame_]                 ;# number of frames
    set Macband [Mac/Tdma set bandwidth_]
    set band [expr 10e6*[string trim $Macband Mb]] ;# bandwidth (b/s)
    set Macslot [Mac/Tdma set max_slot_num_]
    set slots [expr $Macslot + 1]                ;# slots
    set Df [expr $slots*8*$Ls/$band]             ;# frame duration

    # get current time
    set ns_ [simulator instance]
    set now [$ns_ now]

    # evaluate inputs
    set in [$qMonitor_ set barrivals_]
    set out [$qMonitor_ set bdepartures_]
    set drop [$qMonitor_ set bdrops_]

    set buffer [expr $in - $out - $drop]          ;# buffer sample
    set rate [expr $in - $inp]                   ;# rate sample

    if { $rate > $srate } {
        set srate $rate
    } else {
        set srate [expr $rate*$alpha_ + $srate*(1-$alpha_)] ;# smoothed rate
    }

# Pre-Allocator_Support
    if { $active_voice_flow($NodeId) != 0 } {
        if { $active_voice_flow($NodeId) == 1 } {
            set voice_rate 283.974                ;# (VoIP_rate/CRA_rate)*$nf*$Ls
        } else {
            if { $active_voice_flow($NodeId) == 2 } {
                set voice_rate 567.95
            } else {
                if { $active_voice_flow($NodeId) == 3 } {
                    set voice_rate 851.92
                } else {
                    if { $active_voice_flow($NodeId) == 19 } {
                        set voice_rate 3407.69
                    } else {
                        if { $active_voice_flow($NodeId) == 20 } {
                            set voice_rate 3407.69
                        }
                    }
                }
            }
        }
    }

    set srate $voice_rate
}

# Pre-Allocator_Support

```



```

# the request is in slots per frame
set nbuffer [expr ($buffer*$Df)/($Ls*$H_)] ;#[expr $buffer/($Ls*$nf)]
set nsrate [expr $srate/($Ls*$nf)] ;#[expr ($srate*$Df)/($Ls*$H_)]

# calculate request
set req [expr $nsrate*$rbdc + $nbuffer*$vbdc]

# update counters
set inp $in
set tin_ $now

return $req
}

#####
#
# Variable Settings
#
#####

# Skyplex parameters
Mac/Tdma set max_slot_num_ 94 ;# slot per frame (preamble excluded)
Mac/Tdma set slot_packet_len_ 106 ;# slot size (bytes)
Mac/Tdma set num_frame_ 3 ;# multiframe size (in frames)
Mac/Tdma set bandwidth_ 2Mb
Allocator/Proportional set delay_ 1.4
Allocator/Tcl set delay_ 1.4

set EPS 0.000001
set MAX_ACTIVE_NODE 64

LL/Sat set bandwidth_ 2Mb
Simulator instproc get-link {n1 n2} {
    set node1 [$self get-node-by-id $n1]
    return [$node1 set ll_(0)]
}
#

global ns
set ns [new Simulator]

set name satrtpt8_vad

set type RBDC

# Note: Even though "Static" is normally reserved for static
# topologies, the satellite code will trigger a recalculation
# of the routing tables whenever the topology changes.
# Therefore, it is not so much "static" as "omniscient", in that
# topology changes are known instantly throughout the topology.
# See documentation for discussion of dynamic routing protocols.
$ns rtproto Static

#####
# Global configuration parameters
#
#####

global opt
set opt(chan) Channel/Sat
set opt(bw_down) 8Mb; # Downlink bandwidth (satellite to ground)
set opt(bw_up) 2Mb; # Uplink bandwidth
set opt(phy) Phy/Sat
set opt(mac) Mac/Tdma
set opt(ifq) Queue/DropTail
set opt(qlim) 1000
set opt(ll) LL/Sat
set opt(wiredRouting) ON

# IMPORTANT This tracing enabling (trace-all) must precede link and node
# creation. Then following all node, link, and error model
# creation, invoke "$ns trace-all-satlinks $outfile"
set outfile [open $name.tr w]
$ns trace-all $outfile

set ev_file [open $name.out w]

#####
# Set up satellite and terrestrial nodes
#
#####

# GEO satellite: above Brazil -- lets put it at 61 deg. W (Satellite Amazonas)
$ns node-config -satNodeType geo-repeater \
    -phyType Phy/Repeater \

```

```

        -channelType $opt(chan) \
        -downlinkBW $opt(bw_down) \
        -wiredRouting $opt(wiredRouting)

set nodeSat [$ns node]
$nodeSat set-position -61

# Terminals:
$ns node-config -satNodeType terminal
set nodeHub [$ns node]
set nodeTerm(1) [$ns node]
set nodeTerm(2) [$ns node]

$nodeHub set-position -22.9 -43.2;      # Rio de Janeiro
$nodeTerm(1) set-position -1.5 -48.5 ;# Belém
$nodeTerm(2) set-position -19.9 -43.9 ;# Belo Horizonte

#####
# Set up links
#####

# GSLS to the geo satellite:
$nodeHub add-gsl geo $opt(ll) $opt(iff) $opt(qlim) $opt(mac) $opt(bw_up) \
    $opt(phy) [$nodeSat set downlink_] [$nodeSat set uplink_]
set req(0) [$nodeHub install-requester Requester/Tc1]
$nodeHub install-allocator Allocator/Tc1
$nodeHub insert-sat-ttl
$nodeHub trace-event $ev_file

for {set i 1} {$i <= 2} {incr i} {
    $nodeTerm($i) add-gsl geo $opt(ll) $opt(iff) $opt(qlim) $opt(mac) $opt(bw_up) \
        $opt(phy) [$nodeSat set downlink_] [$nodeSat set uplink_]
    set req($i) [$nodeTerm($i) install-requester Requester/Tc1]
    $nodeTerm($i) insert-sat-ttl
}

#####
# Set up Core
#####

$ns node-config -reset
set nodeCore [$ns node]
set nodeMgw1 [$ns node]
set nodeMgw2 [$ns node]
set nodewww [$ns node]

$ns duplex-link $nodeHub $nodeCore 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodeMgw1 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodeMgw2 50Mb 2ms DropTail
$ns duplex-link $nodeCore $nodewww 50Mb 2ms DropTail

#####
# Set up Remote Nodes
#####
set initialnode 9
set numberVSATS 2 ;
for {set lannumber 1} {$lannumber <= $numberVSATS} {incr lannumber} {
    set lan($lannumber) [$ns newLan "" 10Mb 1ms LL Queue/DropTail MAC/802_3 Channel]
    for {set nodenumber [expr $initialnode + 23*($lannumber -1)]} {$nodenumber < [expr
$initialnode + 23*($lannumber -1)] + 22} {incr nodenumber} {
        set n($nodenumber) [$ns node]
        $lan($lannumber) addNode "$n($nodenumber)" 10Mb 1ms
    }
    $lan($lannumber) addNode "$nodeTerm($lannumber)" 10Mb 1ms
}

#####
# Tracing
#####
$ns trace-all-satlinks $outfile
$nodeTerm(1) trace-event $ev_file

#####
# Set up HTTP
#####
# Create page pool as a central page generator. Use PagePool/Math
set pgp [new PagePool/Math]
set tmp [new RandomVariable/Constant] ;## Page size generator
$tmp set val_ 1024 ;## average page size
$pgp ranvar-size $tmp

set tmp [new RandomVariable/Exponential] ;## Page age generator
$tmp set avg_ 5 ;## average page age

```

```

$pgp ranvar-age $tmp

set server [new Http/Server $ns $nodewww] ;## Create a server and link it to the central page
pool
$server set-page-generator $pgp

set cache [new Http/Cache $ns $nodeCore] ;## Create a cache
$cache connect $server

#####
# Set up RTP params
#####

set packetSize 94; # Value in bytes
set ptime 0.04 ; # Value in seconds
set RtpOn 360ms; # Mean time of ON periods
set RtpOff 650ms; # Mean time of OFF periods

#####
# Attach agents
# First Lan
#####

set lannumber 1
# Pre-Allocator_Support
set VSatIndex [expr $lannumber + 1] ;# id from node $nodeTerm(1)
# Pre-Allocator_Support

set nodenumber [expr $initialnode + 23*($lannumber -1)]

set s($nodenumber) [new Agent/UDP]
set Null($nodenumber) [new Agent/Null]
$ns attach-agent $n($nodenumber) $s($nodenumber)
$ns attach-agent $nodeMgw1 $Null($nodenumber)
$ns connect $s($nodenumber) $Null($nodenumber)
set Exp($nodenumber) [new Application/Traffic/Exponential]
$Exp($nodenumber) attach-agent $s($nodenumber)
$Exp($nodenumber) set packetSize_ $packetSize
$Exp($nodenumber) set burst_time_ $RtpOn
$Exp($nodenumber) set idle_time_ $RtpOff
$Exp($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
set f($nodenumber) [new Agent/UDP]
set Nullf($nodenumber) [new Agent/Null]
$ns attach-agent $nodeMgw1 $f($nodenumber)
$ns attach-agent $n($nodenumber) $Nullf($nodenumber)
$ns connect $f($nodenumber) $Nullf($nodenumber)
set Expf($nodenumber) [new Application/Traffic/Exponential]
$Expf($nodenumber) attach-agent $f($nodenumber)
$Expf($nodenumber) set packetSize_ $packetSize
$Expf($nodenumber) set burst_time_ $RtpOn
$Expf($nodenumber) set idle_time_ $RtpOff
$Expf($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
$ns at 24 "$Exp($nodenumber) start"
$ns at 24 "$Expf($nodenumber) start"
# Pre-Allocator_Support
$ns at 24 "set active_voice_flow($VSatIndex) 20"
# Pre-Allocator_Support
$ns at 80 "$Exp($nodenumber) stop"
$ns at 80 "$Expf($nodenumber) stop"
# Pre-Allocator_Support
$ns at 80 "set active_voice_flow($VSatIndex) 19"
# Pre-Allocator_Support

for {set nodecounter 1} {$nodecounter <= 19} {incr nodecounter} {

set secondnode [expr $nodenumber + $nodecounter]
set s($secondnode) [new Agent/UDP]
set Null($secondnode) [new Agent/Null]
$ns attach-agent $n($secondnode) $s($secondnode)
$ns attach-agent $nodeMgw2 $Null($secondnode)
$ns connect $s($secondnode) $Null($secondnode)
set Exp($secondnode) [new Application/Traffic/Exponential]
$Exp($secondnode) attach-agent $s($secondnode)
$Exp($secondnode) set packetSize_ $packetSize
$Exp($secondnode) set burst_time_ $RtpOn
$Exp($secondnode) set idle_time_ $RtpOff
$Exp($secondnode) set rate_ [expr 8*$packetSize/$ptime]
set f($secondnode) [new Agent/UDP]
set Nullf($secondnode) [new Agent/Null]
$ns attach-agent $nodeMgw2 $f($secondnode)
$ns attach-agent $n($secondnode) $Nullf($secondnode)
$ns connect $f($secondnode) $Nullf($secondnode)
set Expf($secondnode) [new Application/Traffic/Exponential]

```

```

    $Expf($secondnode) attach-agent $f($secondnode)
    $Expf($secondnode) set packetSize_ $packetSize
    $Expf($secondnode) set burst_time_ $RtpOn
    $Expf($secondnode) set idle_time_ $RtpOff
    $Expf($secondnode) set rate_ [expr 8*$packetSize/$ptime]
    $ns at [expr 0.3 + ($nodecounter - 1)] "$Exp($secondnode) start"
    $ns at [expr 0.3 + ($nodecounter - 1)] "$Expf($secondnode) start"
# Pre-Allocator_Support
    $ns at 14.3 "set active_voice_flow($VSatIndex) 19"
# Pre-Allocator_Support
    $ns at 80 "$Exp($secondnode) stop"
    $ns at 80 "$Expf($secondnode) stop"
# Pre-Allocator_Support
    $ns at 80.02 "set active_voice_flow($VSatIndex) 0"
# Pre-Allocator_Support
}

set thirdnode [expr $nodenumber + 20]
set client($lannumber) [new Http/Client $ns $n($thirdnode)] ;## Create a client
set tmp [new RandomVariable/Exponential] ;## Poisson process as request sequence
$tmp set avg_ 5 ;## average request interval
$client($lannumber) set-interval-generator $tmp
$client($lannumber) set-page-generator $pgp
$client($lannumber) connect $cache
$ns at 0.0 "$client($lannumber) start-session $cache $server"

set fourthnode [expr $nodenumber + 21]
set tcp($lannumber) [new Agent/TCP]
set tcpsink($lannumber) [new Agent/TCPSink]
$ns attach-agent $nodewww $tcp($lannumber)
$ns attach-agent $n($fourthnode) $tcpsink($lannumber)
$ns connect $tcp($lannumber) $tcpsink($lannumber)
set ftp($lannumber) [new Application/FTP]
$ftp($lannumber) attach-agent $tcp($lannumber)
$ns at 0 "$ftp($lannumber) start"
$ns at 80 "$ftp($lannumber) stop"

#####
# Attach agents
# Subsequent Lans
#####

for {set lannumber 2} {$lannumber <= $numberVSats} {incr lannumber} {
# Pre-Allocator_Support
    set VSatIndex [expr $lannumber + 1]
# Pre-Allocator_Support
    set nodenumber [expr $initialnode + 23*($lannumber -1)]

    set s($nodenumber) [new Agent/UDP]
    set Null($nodenumber) [new Agent/Null]
    $ns attach-agent $n($nodenumber) $s($nodenumber)
    $ns attach-agent $nodeMgw2 $Null($nodenumber)
    $ns connect $s($nodenumber) $Null($nodenumber)
    set Exp($nodenumber) [new Application/Traffic/Exponential]
    $Exp($nodenumber) attach-agent $s($nodenumber)
    $Exp($nodenumber) set packetSize_ $packetSize
    $Exp($nodenumber) set burst_time_ $RtpOn
    $Exp($nodenumber) set idle_time_ $RtpOff
    $Exp($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
    set f($nodenumber) [new Agent/UDP]
    set Nullf($nodenumber) [new Agent/Null]
    $ns attach-agent $nodeMgw2 $f($nodenumber)
    $ns attach-agent $n($nodenumber) $Nullf($nodenumber)
    $ns connect $f($nodenumber) $Nullf($nodenumber)
    set Expf($nodenumber) [new Application/Traffic/Exponential]
    $Expf($nodenumber) attach-agent $f($nodenumber)
    $Expf($nodenumber) set packetSize_ $packetSize
    $Expf($nodenumber) set burst_time_ $RtpOn
    $Expf($nodenumber) set idle_time_ $RtpOff
    $Expf($nodenumber) set rate_ [expr 8*$packetSize/$ptime]
    $ns at [expr 0.1 + ($lannumber - 1)] "$Exp($nodenumber) start"
    $ns at [expr 0.1 + ($lannumber - 1)] "$Expf($nodenumber) start"
# Pre-Allocator_Support
    $ns at 1.1 "set active_voice_flow($VSatIndex) 1"
# Pre-Allocator_Support
    $ns at 80 "$Exp($nodenumber) stop"
    $ns at 80 "$Expf($nodenumber) stop"
# Pre-Allocator_Support
    $ns at 80 "set active_voice_flow($VSatIndex) 19"
# Pre-Allocator_Support
}

```

```

for {set nodecounter 1} {$nodecounter <= 19} {incr nodecounter} {
    set secondnode [expr $nodenumber + 1]
    set s($secondnode) [new Agent/UDP]
    set Null($secondnode) [new Agent/Null]
    $ns attach-agent $n($secondnode) $s($secondnode)
    $ns attach-agent $nodeMgw2 $Null($secondnode)
    $ns connect $s($secondnode) $Null($secondnode)
    set Exp($secondnode) [new Application/Traffic/Exponential]
    $Exp($secondnode) attach-agent $s($secondnode)
    $Exp($secondnode) set packetSize_ $packetSize
    $Exp($secondnode) set burst_time_ $RtpOn
    $Exp($secondnode) set idle_time_ $RtpOff
    $Exp($secondnode) set rate_ [expr 8*$packetSize/$ptime]
    set f($secondnode) [new Agent/UDP]
    set Nullf($secondnode) [new Agent/Null]
    $ns attach-agent $nodeMgw2 $f($secondnode)
    $ns attach-agent $n($secondnode) $Nullf($secondnode)
    $ns connect $f($secondnode) $Nullf($secondnode)
    set Expf($secondnode) [new Application/Traffic/Exponential]
    $Expf($secondnode) attach-agent $f($secondnode)
    $Expf($secondnode) set packetSize_ $packetSize
    $Expf($secondnode) set burst_time_ $RtpOn
    $Expf($secondnode) set idle_time_ $RtpOff
    $Expf($secondnode) set rate_ [expr 8*$packetSize/$ptime]
    $ns at [expr 0.1 + ($nodecounter - 1)] "$Exp($secondnode) start"
    $ns at [expr 0.1 + ($nodecounter - 1)] "$Expf($secondnode) start"
# Pre-Allocator_Support
    $ns at 14.1 "set active_voice_flow($VSatIndex) 19"
# Pre-Allocator_Support
    $ns at 80 "$Exp($secondnode) stop"
    $ns at 80 "$Expf($secondnode) stop"
# Pre-Allocator_Support
    $ns at 80.02 "set active_voice_flow($VSatIndex) 0"
# Pre-Allocator_Support

}

set thirdnode [expr $nodenumber + 20]
set client($lannumber) [new Http/Client $ns $n($thirdnode)] ;## Create a client
set tmp [new RandomVariable/Exponential] ;## Poisson process as request sequence
$tmp set avg_ 5 ;## average request interval
$client($lannumber) set-interval-generator $tmp
$client($lannumber) set-page-generator $pgp
$client($lannumber) connect $cache
$ns at 0.0 "$client($lannumber) start-session $cache $server"

set fourthnode [expr $nodenumber + 21]
set tcp($lannumber) [new Agent/TCP]
set tcpsink($lannumber) [new Agent/TCPSink]
$ns attach-agent $nodewww $tcp($lannumber)
$ns attach-agent $n($fourthnode) $tcpsink($lannumber)
$ns connect $tcp($lannumber) $tcpsink($lannumber)
set ftp($lannumber) [new Application/FTP]
$ftp($lannumber) attach-agent $tcp($lannumber)
$ns at 0 "$ftp($lannumber) start"
$ns at 80 "$ftp($lannumber) stop"
}

#####
# Satellite routing
#####

set satrouteobject_ [new SatRouteObject]
$satrouteobject_ compute_routes

$ns at 90 "finish"

proc finish {} {
    global ns outfile
    $ns flush-trace
    close $outfile
    $ns halt
}

$ns run

```